UNIVERSIDADE DE SÃO PAULO
ESCOLA DE COMUNICAÇÃO E ARTES
PROGRAMA DE PÓS-GRADUAÇÃO EM MÚSICA

JOÃO PAULO SILVA MONNAZZI

***Body Building Music: sistema interativo para
criação musical controlado por *performance*
coreográfica esportiva***

São Paulo
2018

***Body Building Music: sistema interativo para criação musical
controlado por performance coreográfica esportiva***

Dissertação apresentada ao Departamento de
Música da Escola de Comunicação e Artes de
São Paulo da Universidade de São Paulo para
obtenção do título de Mestre em Música.

Área de Concentração: Processos de Criação
Musical

Orientador: Prof. Dr. Regis Rossi Alves Faria

São Paulo

2018

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Catálogo na Publicação
Serviço de Biblioteca e Documentação
Escola de Comunicações e Artes da Universidade de São Paulo
Dados inseridos pelo(a) autor(a)

Monnazzi, João Paulo Silva
Body Building Music: sistema interativo para criação musical controlado por performance coreográfica esportiva / João Paulo Silva Monnazzi ; orientador, Regis Rossi Alves Faria. -- São Paulo, 2018.
127 p.: il.

Dissertação (Mestrado) - Programa de Pós-Graduação em Música - Escola de Comunicações e Artes / Universidade de São Paulo.
Bibliografia
Versão corrigida

1. Criação musical interativa 2. sensoriamento de gestos corporais 3. computação musical 4. Fisiculturismo 5. Kinect
I. Faria, Regis Rossi Alves II. Título.

CDD 21.ed. - 780

Elaborado por Sarah Lorenzon Ferreira - CRB-8/6888

João Paulo Silva Monnazzi

Título: *Body Building Music*: sistema interativo para criação musical controlada por *performance* coreográfica esportiva

Dissertação apresentada ao Departamento de
Música da Escola de Comunicação e Artes de
São Paulo da Universidade de São Paulo para
obtenção do título de Mestre em Música.

Área de Concentração: Processos de Criação
Musical

Aprovado em:

Banca Examinadora

Prof. Dr. _____

Instituição: _____ Assinatura: _____

Prof. Dr. _____

Instituição: _____ Assinatura: _____

Prof. Dr. _____

Instituição: _____ Assinatura: _____

Dedico esta Dissertação a meus amores, José e Maria Monnazzi.

AGRADECIMENTOS

Ao meu amigo e orientador, Prof. Dr. Regis Rossi Alves Faria, por acreditar neste trabalho. Por todo incentivo, conselhos, pelas sugestões e pela oportunidade de contínuo aprendizado e desenvolvimento científico ao seu lado.

Ao meu sobrinho, Carlos Gabriel Luz Monnazzi, por toda dedicação, trabalho e companheirismo durante a programação da parte prática do sistema.

Ao meus pais, Maria e José Monnazzi, pelo amor e apoio incondicionais de uma vida, sendo exemplos de honestidade, dedicação, trabalho e entrega para o bem maior.

Aos membros titulares da Comissão Julgadora, Almir Almas e Anselmo Guerra, pela atenção, simpatia e disponibilidade em avaliar este trabalho, conferindo e valorizando o mérito científico de todo o nosso esforço.

Aos meus irmãos, José Carlos, Luis Gustavo e Marcelo, por serem fonte de inspiração e exemplo de dedicação e profissionalismo em suas jornadas.

Ao Bruno Mengatti e família pelas décadas de amizade, companheirismo e acolhida nos últimos 4 anos de idas e vindas para São Paulo.

À CAPES, pelo apoio financeiro concedido.

Aos colegas da Pós-Graduação, pela amizade e pelas experiências compartilhadas.

Ao Prof. Dr. Rodolfo Coelho de Souza e a Prof. Dr. Patrícia Moran, pela revisão e conselhos durante a etapa de qualificação.

Ao Guilherme Feulo, pelo auxílio na montagem dos *patches* do programa Pure Data.

Ao Guilherme Paviani, pelo auxílio na gravação dos vídeos de demonstração do sistema.

A minha namorada, Angélica Bertão, pelo apoio, carinho, estímulo, ajuda, conselhos, compreensão, paciência e por tudo aquilo mais que eventualmente eu tenha esquecido de mencionar aqui.

A ESTALAGEM DA RAZÃO

A meio caminho entre a fé e a crítica está a estalagem da razão. A razão é a fé no que se pode compreender sem fé; mas é uma fé ainda, porque compreender envolve pressupor que há qualquer coisa compreensível.

– Trecho 176

Fernando Pessoa – Livro do Desassossego

Este trabalho contou com o seguinte apoio financeiro:

- Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)

RESUMO

Em meio a diversas interações midiáticas oriundas do desenvolvimento tecnológico no século XXI, percebemos que a música pode desbravar novos caminhos e conexões em áreas ainda pouco exploradas, como na interdisciplinaridade com o esporte. Este trabalho tem como foco a concepção de um sistema de criação musical interativo controlado por *performance* coreográfica esportiva, onde o atleta, além de interagir com o processo criativo e com a própria música durante sua performance coreográfica, possa também alterar sua estrutura e redefinir parâmetros de condução e geração sonora. Através de um sensor de movimento o atleta atua em tempo real reorganizando a execução de materiais sonoros previamente elaborados e manipulando processadores de efeito e de dinâmica. Uma composição modular interativa foi realizada para testar o sistema, adotando-se uma estrutura musical modular em tonalidades. Atribuindo ao atleta funções antes exclusivas do nicho profissional do intérprete e/ou compositor musical, o sistema desenvolvido oferece o acesso a processos de criação musical a uma população que não necessariamente detém conhecimento sobre as ferramentas tradicionais de composição musical. Os resultados exploram novas formas de mediações e abrem fronteiras de aplicações intocadas, bem como novos nichos de trabalho para o profissional da música nas áreas de criação, produção e performance.

Palavras-chave: Criação Musical Interativa; Sensoriamento de Gestos Corporais; Kinect; Performance Esportiva; Interação em Tempo Real; Computação Musical; Fisiculturismo.

ABSTRACT

In the midst of several media interactions that came from technological development in the 21st century, we realized that music can break new ground and connections in areas that are still little explored, such as in interdisciplinarity with sports. This work focuses on the conception of an interactive musical creation system controlled by a choreographic performance, in which the athlete, in addition to interacting with the creative process and with the music itself during its choreographic performance, can also alter its structure and redefine parameters of conduction and sound generation. Through a motion sensor the athlete acts in real time reorganizing the execution of previously elaborated sound materials and manipulating processors of effect and dynamics. An interactive modular composition was performed to test the system, adopting a modular musical structure in shades. By providing the athlete with functions previously unique to the professional niche of the performer and / or music composer, the developed system offers access to music creation processes for a population that does not necessarily have knowledge about traditional musical composition tools. The results explore new forms of mediation and open boundaries of untouched applications, as well as new work niches for the music professional in the areas of creation, production and performance.

Keywords: Interactive Musical Creation; Body Sensor Gesture, Kinect; Sportive Performance; Live Interaction; Computer Music; Bodybuilding

LISTA DE ABREVIATURAS E SIGLAS

ACM	<i>Association for Computing Machinery</i>
AEC	Abdominal e Coxa
AES	<i>Audio Engineering Society</i>
BBM	<i>Body Building Music</i>
BBMC	<i>Body Building Music Controller</i>
dB	decibel
DAW	<i>Digital Audio Workstation</i>
DBC	Duplo Bíceps de Costas
DBF	Duplo Bíceps de Frente
EDC	Expansão de Dorsais de Costas
EDF	Expansão de Dorsais de Frente
EG	<i>Eurographics Association</i>
Fc	Frequência Central
Hz	Hertz
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IFBB	<i>International Federation of Bodybuilding</i>
IMD	Instrumento Musical Digital
MDA	Mão Direita Aberta
MEA	Mão Esquerda Aberta
MHA	Mão Horizontal Aberta
MHF	Mão Horizontal Fechada
MM	Mais Musculosa
MVA	Mão Vertical Alto

MVB	Mão Vertical Baixo
NUI	<i>Natural User Interface</i>
OSC	<i>Open Sound Control</i>
PD	Pure Data
PML	Peitoral Melhor Lado
Q	Fator de qualidade
SDK	<i>System Development Kit</i>
TML	Tríceps Melhor Lado
ToF	<i>Time of Flight</i>
USP	Universidade de São Paulo

LISTA DE FIGURAS

Figura 1: Estátua de Hércules de Farnese (esquerda) e Laocoon (direita).....	30
Figura 2: Homem Vitruviano, desenhado por Leonardo Da Vinci	31
Figura 3: Eugen Sandow em uma apresentação no início do séc. XX.....	33
Figura 4: Estrutura de hardware e software da peça Vertimas - übersetzen (KRONLACHNER, 2012)	39
Figura 5: Diagrama de interação MotionComposer (BERGLAND et al, 2015)	40
Figura 6: Fluxograma do sistema interativo MindSounDS (TAVARES et al, 2015).....	43
Figura 7: Diagrama de processo do ambiente Kinectofon (JENSENIUS, 2013)	45
Figura 8: Fluxograma do sistema interativo Crosssole (SENTURK et al, 2012).....	47
Figura 9: Diagrama de conexão dos hardwares constituintes do sistema BBM	57
Figura 10: Fluxograma do sistema BBM.	58
Figura 11: Comparação dos modelos Kinect 1 e 2. (PAGLIARI et al, 2015)	59
Figura 12: Representação dos <i>outputs</i> de dados do Kinect V2 (LACHAT et al, 2015)	61
Figura 13: Aba de leitura de vídeo do Kinect Studio.....	64
Figura 14: Imagem captada em <i>colored cloud point</i> pelo sensor Kinect.....	65
Figura 15: Opções de <i>streams</i> selecionadas na captura de vídeo das poses do sistema BBM.	65
Figura 16: Uma das quatro pastas de arquivos com vídeos gravados pelo Kinect Studio para exemplificar o tamanho dos arquivos de imagem.....	66
Figura 17: Marcação das poses discretas no VGB para a construção do repositório	69
Figura 18: Repositórios criados no VGB para o banco de poses dos níveis 1,2 e 3	69
Figura 19: Marcação do gesto discreto inicial do movimento	70
Figura 20: Marcação do gesto discreto final do movimento.....	71
Figura 21: Marcação do gesto contínuo entre o ponto de início e de término.....	71
Figura 22: Programação da ferramenta <i>Kinect_Gesture to OSC</i> no Microsoft Visual Studio	73
Figura 23: Programa de iniciação do middleware do sistema BBM.....	75
Figura 24: Código C# da seção de reconhecimento de gestos do <i>Kinect_Gesture to OSC</i> no Microsoft Visual Studio	82
Figura 25: Código C# do módulo Transmissor OSC, que perfaz a programação de comunicação para envio de mensagens entre o Kinect e o Pure Data através do protocolo OSC.....	85
Figura 26: Código C# do módulo Translator, que perfaz a configuração dos parâmetros de comparação entre o banco de poses e os dados captados pelo Kinect	89
Figura 27: Fluxograma do gerador e controlador musical BBMC	90

Figura 28: <i>Patch</i> do gerador musical BBMC.....	91
Figura 29: Parte do patch responsável pelo recebimento dos dados através do protocolo OSC	92
Figura 30: <i>Subpatch</i> responsável pela sincronização dos objetos musicais.....	93
Figura 31: Módulo responsável pelo controle de disparo dos objetos sonoros	94
Figura 32: <i>subpatch</i> responsável pelo efeito de filtro	95
Figura 33: Módulo responsável pelo controle de dinâmica	96
Figura 34: Sessão da composição Kinase na DAW Logic Pro X	97
Figura 35: Estruturação da composição interativa	98
Figura 36: <i>print screen</i> da peça audiovisual Kinase	102

LISTA DE TABELAS

Tabela 1 - Comparação de aspectos técnicos dos projetos	48
Tabela 2: Mapeamento de poses do nível 1 - Controle Musical	54
Tabela 3: Mapeamento de poses do Nível 2 - Controle de Efeito	55
Tabela 4: Mapeamento de poses Nível 3 - Controle de Volume	55

SUMÁRIO

1. INTRODUÇÃO	23
1.1 Justificativas	25
1.2 Objetivos	27
1.3 Organização da dissertação	27
2. <i>In Kinesis</i> : O digital como interlocutor entre esporte e música	29
2.1 Fisiculturismo: A construção do corpo	29
2.1.1 Da subcultura às Olimpíadas.....	31
2.1.2 Em busca do corpo perfeito: Competições de Fisiculturismo.....	34
2.2 Instrumentos Musicais Digitais (IMDs).....	36
2.2.2. Sistemas gerativos de Música através de Interfaces de Usuário Naturais	38
3. Projeto de um sistema interativo musical controlado por performance coreográfica utilizando o <i>Kinect</i>	50
3.1 Concepção e desenvolvimento do sistema.....	51
3.1.1 Requerimentos do Sistema.....	56
3.1.2. <i>Kinect</i>	58
3.2. Construindo o banco de poses.....	63
3.2.1 Captação do banco de poses.....	63
3.2.2 Construção de uma biblioteca de poses a partir de gestos captados	67
3.3. Programação do <i>Middleware</i> : transcritor coreográfico-musical e montador de mensagens OSC.....	72
3.3.1 – Classe de iniciação do middleware	74
3.3.2 <i>Gesture Detector</i>	75
3.3.3 Transmissor OSC	82
3.3.4 Translator	85
3.4. O módulo gerador e controlador musical.....	89
3.4.1 Módulo receptor e distribuidor de mensagens OSC	91
3.4.2 Máquina do Tempo (sincronizador de disparo de objetos sonoros)	92
3.4.3 Operador de nível 1: Executor de objetos sonoros	93
3.4.4 Operador de nível 2: Processador de efeitos	94
3.4.5 Operador de Nível 3: Processador dinâmico.....	95
3.5. <i>Kinase</i> : uma composição modular interativa para BBM	96
3.5.1. A Estrutura Musical de <i>Kinase</i>	98

3.6 Sequencias de operações para o uso do sistema BBM.....	101
4. Conclusões	102
4.1 Resultados	102
4.2 Análise e discussão sobre os resultados	105
4.3 Considerações Finais.....	108
REFERÊNCIAS BIBLIOGRÁFICAS.....	110
APÊNDICE A – Partitura de Kinase: composição dinâmica para sistema musical interativo	117
ANEXO A – <i>Datasheet</i> do requerimento da configuração necessário para utilizar o Kinect sensor V2.....	127
ANEXO B – Diretrizes normativas das poses mandatórias para categoria fisiculturismo masculino	131

1. INTRODUÇÃO

Desde o advento do fonógrafo em 1877 a interação com a música nunca mais foi a mesma. A rápida explosão tecnológica do século XX trouxe consigo mudanças comportamentais nos mais variados âmbitos da sociedade. Ross (2010) inclusive nos indaga sobre os benefícios e malefícios que o processo de gravação trouxe para relações sociais e culturais assim como seu impacto na história da música.

Um benefício indiscutível foi a disseminação em uma escala jamais vista de material sonoro. Com o decorrer das décadas a manipulação de áudio foi se tornando cada vez mais prática e acessível ao público de um modo geral, muitas vezes mesmo sem os conhecimentos básicos de como tocar, editar ou compor música. Henriques (2008) nos lembra que a evolução tecnológica foi tão grande que não fazemos ideia do quanto de tecnologia estamos em contato quando acessamos nossa *DAW (Digital Audio Workstation)* e acionamos nosso controlador.

A possibilidade de armazenamento digital de áudio proporcionou um aumento expressivo das possibilidades de sua manipulação, criando várias áreas de atuação novas dentro do meio musical e também fora dela, uma que pode ser citada é a interatividade musical. Salter (2010) discute o impacto da tecnologia e a mudança da experiência musical, como os instrumentos digitais e os processos de manipulação de áudio através do computador pessoal desconstruíram as linhas que delimitavam quem produz e quem consome, entre intérprete e plateia, diluindo as margens que separa o produtor do som do receptor, tirando este do mero papel de ouvinte e o levando para uma zona interativa de inúmeras possibilidades de controles musicais que antes eram de domínio exclusivo de músicos capacitados. No entanto, hoje, as ferramentas de criação sobrepõem a necessidade do conhecimento teórico, musical e tecnológico aprofundado.

Segundo Iazzetta (2009) a função de tornar musical a tecnologia eletrônica não é tarefa fácil pois deve-se lidar com as limitações dessa tecnologia assim como com o desafio de tornar coerente a interação musical. Faz-se necessário a concepção de conexões, seja no campo teórico ou prático, entre os assuntos abordados que justifiquem a montagem do sistema.

Para a montagem do sistema *Body Building Music*, a primeira ponte que podemos observar é a influência psicológica que a música pode ter quando ouvida durante a prática

de atividades físicas. Wijnalda (2005) ressalta que a música, através de associação e motivação, pode aumentar o desempenho de um atleta durante a atividade física, sendo que estímulos externos podem reduzir a percepção de esforço assim como remontar a sensações psicológicas de recompensa. Uma pesquisa desenvolvida por Silva e Gress (2012) aponta que em torno de 96,5% de praticantes de musculação escutam música durante o treino.

Segundo Bordieu (1983), no período da Revolução Industrial ao final do século XIX, o esporte foi implementado nas escolas da Inglaterra com o objetivo de transmitir valores aristocráticos de distinção, cavalheirismo, *fair play* e diminuição da violência. Esta abordagem evidencia outro ponto em comum entre as áreas de esporte e arte. A função de transmitir valores universais necessários para um melhor convívio em sociedade.

Este trabalho escolhe o esporte *bodybuilding* – fisiculturismo, em uma tradução conveniente – para uma inédita exploração experimental interativa com a criação musical. Dentre várias razões pela escolha inclui-se por ser esta uma das práticas esportivas que, dentro da competição, contém uma apresentação coreográfica, a qual é julgada sob critérios estéticos artísticos tais como ritmo, sincronismo e adequação da coreografia com a trilha sonora escolhida.

O fisiculturismo, segundo Locks (2012), não pode ser visto através de uma análise simplista sobre os conceitos básicos do esporte, como treinamento resistido com pesos e a manutenção de dietas para manipulação da forma física. Estas são ferramentas na mão do atleta que, através desses, busca o utópico ideal do corpo perfeito, que observamos não ser uma preocupação meramente contemporânea. Estátuas gregas como as de Hércules (Hércules) e Apolo construídas na Grécia antiga nos mostram que o interesse pelo físico atlético e seu estudo datam de muito tempo (Bittencourt *et al apud* Neves, 2012).

O fisiculturismo acaba de se tornar esporte olímpico e tem sua estreia programada para acontecer nos jogos Pan Americanos de 2019, em Lima no Peru. Uma das partes mais aclamadas do evento são as apresentações individuais dos atletas, onde eles devem posar com uma música de fundo, respeitando parâmetros musicais como tempo, andamento e cadência. Ao se abrir um potencial de novas aplicações musicais direcionadas a este novo esporte olímpico, consideramos apropriado abordar algumas perspectivas deste com a música.

Neste contexto, este trabalho busca investigar como estreitar a relação entre apresentação esportiva e o fazer da música, vislumbrando o atleta como um compositor e intérprete, transformando seu corpo em um instrumento musical controlador, e tornando a experiência audiovisual da apresentação ainda mais interativa.

1.1 Justificativas

Em um primeiro momento, sob um olhar ligeiro e menos interessado, podemos não perceber relações entre arte e esporte de maneira tão clara como a civilização grega de outrora. Platão¹, não por acaso, já salientava que para a formação completa do guerreiro, a música, para uma alma sóbria, e a ginástica, para um corpo sadio, são necessárias. Mas o passar dos séculos e o aprimoramento das especificidades de cada atividade nos legou uma (ilusória) sensação de que músicos e atletas são seres de mundos diferentes, com a comunicação delimitada a apreciação entre espectador e ator.

Ao transitar entre âmbitos aparentemente distantes, é o foco principal deste trabalho pesquisar como expandir as possibilidades de interações musicais na performance esportiva, e fomentar a criação musical mediada pela tecnologia a um novo nicho, onde os papéis de compositor e atleta se tornam indissociáveis, com cada ator e sua retórica influenciando diretamente todo o processo de geração musical.

Neste cenário, o compositor se vê diretamente ligado à programação do sistema, assim como na escolha dos movimentos executados pelo atleta, enquanto ao mesmo tempo o atleta se torna compositor ao ganhar a possibilidade de alterar materiais sonoros durante sua apresentação, tornando-os células de um mesmo organismo.

Dentro do contexto sociocultural brasileiro, a capacidade da música em se adequar a diferentes propósitos e servir como meio catalisador para diversas formas de discursos ao se encontrar com as modalidades esportivas, cria uma nova via de comunicação entre música e esporte, mediada pela tecnologia. O projeto almeja estreitar este laço multidisciplinar, contemplando criar novas oportunidades profissionais para ambas as

¹ República, III, 404d-405d

áreas culturais, que são identidades importantes no nosso país e detém uma grande função na formação da estética social brasileira.

Buscando a formatação de um sistema de controle de geração de sons musicais via leitura do movimento (gestual), os resultados deste trabalho poderão servir como ferramenta para a criação de peças musicais interativas, assim como servirá de provocação para a busca de novas áreas de pesquisa a serem tangenciadas por práticas musicais. Poderão também suscitar novas formas de interação em áreas onde a música já assume um papel fundamental, como no teatro, apresentações de dança, instalações artísticas, apresentações esportivas como o nado sincronizado, ginástica rítmica dentre outras modalidades.

Interações mediadas através de sensores gestuais já são utilizadas em diversos tipos de aplicações, sendo que desenvolvedores de *softwares* tem dado cada vez mais importância a estudos relacionados. O principal motivo é tornar a interação entre homem e máquina cada vez mais natural utilizando o repertório de gestos desenvolvidos ao longo dos anos por nossa espécie como controles de comandos virtuais (Corrêa apud Silva et al, 2013). Um exemplo que podemos citar é do pioneiro David Rokeby. Em sua instalação artística denominada *Very Nervous System* (ROKEBY, 1995) a *performance* é quem comanda o sistema, mapeando informação visual em sons.). Neste sistema capta-se a imagem dos movimentos do corpo de uma pessoa através de câmeras e a partir dessa sinalização controlam-se elementos de síntese sonora.

Com a transdisciplinaridade, é capaz de fomentar a sinergia entre esporte e música por intermédio da tecnologia durante uma apresentação, Salter (2010) chega a citar a tecnologia como o Santo Graal da performance, o meio capaz de convergir as mais diversas áreas do conhecimento e criar diálogos que antes não existiriam.

A criação da ferramenta conceberá um diálogo mais próximo entre áreas que já se relacionam, fornecendo a possibilidade de novas discussões na área desportiva e artística.

Ao trabalhar com as duas áreas de forma intensa, porém separadamente, observa-se a janela de oportunidade que existe para intermediações, pois assim como há algo de artístico em todo esporte, há algo atlético em toda performance artística (GUMBRECHT, 2007), de forma que consideramos como uma das inovações nesta proposta o objetivo de aproximar a área de aplicação musical com a prática esportiva, duas áreas com pouca interação e/ou interdisciplinaridade explorada até o momento.

1.2 Objetivos

Esta pesquisa procura analisar as possibilidades e meios de produção de música interativa controlada via leitura de movimento corpóreo com particular interesse no gestual de natureza esportiva. São objetivos desta pesquisa:

- Levantar e definir estratégias de mapeamento entre gestual esportivo e materiais sonoros musicais, que façam sentido tanto no contexto do esporte quanto no contexto musical, para fins de composição ou produção de música concomitante à prática esportiva
- Conceber um sistema interativo para geração musical, constituído de sensores e processadores para disparo e controle de elementos musicais em tempo real durante uma *performance* coreográfica de fisiculturismo.
- Realização de um estudo de caso de criação/geração de música interativa controlada/mediada por uma *performance* coreográfica esportiva, usando-se a técnica de *sequenciamento horizontal* (MENEGUETTE, 2011), modelo básico do conceito de música adaptativa e o mapeamento gesto-controle definido para o caso particular de uma *performance* de fisiculturismo.
- Fomentar discussão e análise das possibilidades de emprego das técnicas aqui apresentadas, e desenvolvimentos futuros dentro da linha de pesquisa de criação musical controlada por gestuais oriundos de práticas esportivas.

1.3 Organização da dissertação

No capítulo 1 apresenta-se uma breve introdução aos avanços tecnológicos na área do áudio que possibilitaram a implementação do sistema proposto neste trabalho. Também é apresentado o conteúdo que justifica o escopo do tema, as razões para realização da pesquisa e a motivação para a concepção e montagem de um sistema musical interativo controlado por *performance* coreográfica esportiva.

No capítulo 2 apresenta-se uma contextualização teórica individualizada das áreas que são abordadas pelo projeto. Neste capítulo busca-se esclarecer os objetos da pesquisa,

expondo um referencial teórico para que o leitor compreenda os dois âmbitos que o sistema intersecciona. Na primeira parte deste capítulo temos a contextualização da modalidade esportiva escolhida para estudo de caso: o fisiculturismo, constituída por uma breve contextualização histórica e social, e pela explicação dos parâmetros que regem os campeonatos, tendo em vista que o sistema foi montado observando as normas descritas no regulamento que rege as competições oficiais da Federação Internacional de *BodyBuilding* (IFBB) .

A segunda parte do capítulo 2 se volta à apresentação da categoria de instrumentos musicais digitais em que o sistema proposto se enquadra, devido a sua forma de manipulação e ao mapeamento entre gesto e controle. Essa parte termina com um levantamento bibliográfico de projetos que utilizaram o mesmo sensor (Kinect) utilizado neste projeto, comparando aspectos de montagem, implementação, mapeamento e controle de cada projeto. Este referencial serviu como base para o desenvolvimento deste projeto, dando respaldo teórico aos materiais e métodos escolhidos para realização do trabalho.

O capítulo 3 trata da implementação do sistema em si, expondo os materiais e métodos utilizados no processo, e percorrendo sobre cada etapa da montagem. A abordagem cobre da concepção e estruturação da ideia do projeto ao processo criativo da composição musical, passando pela programação dos *softwares* e montagem dos *hardwares*.

No capítulo 4 apresentam-se os resultados, e discussões onde avalia-se o sistema em seu primeiro estado (protótipo) e apresentam-se possibilidades para implementações futuras, assim como outras possibilidades de uso do sistema interativo.

2. *In Kinesis*: O digital como interlocutor entre esporte e música

Kinesis, palavra oriunda da língua grega que designa movimento. Movimento, que nesta dissertação, se torna fator comum em escopos aparentemente afastados, conectando-os pela presença. Presença do corpo, do corpo em ação, do corpo criando diálogo, mediante a *performance*, através das linguagens pré-estabelecidas dentro de um determinado nicho cultural.

Arte e esporte comungam sob as mesmas competências práticas necessárias por parte de seus praticantes. Não há como se tornar um bom intérprete sem anos de prática assim como o atleta. Assim como vemos lances esportivos que não há outra forma de explica-lo a não ser como uma “obra de arte”.

Salter (2010) defende que um dos pontos positivos da evolução tecnológica é a capacidade de conexão ou estreitamento entre áreas pouco correlatas que se torna possível quando há a intermediação digital.

Neste capítulo apresentaremos os temas que são transpassados pelo sistema interativo concebido.

2.1 Fisiculturismo: A construção do corpo

No final do século XIX, um novo interesse pelo fisiculturismo surgiu, não pelos músculos simplesmente como meio de sobrevivência ou autodefesa; houve um retorno do ideal grego, o desenvolvimento muscular como celebração do corpo humano (SCHWARZENEGGER, 1998, p.30)

A etimologia da palavra fisiculturismo vem da junção do radical grego PHYSIOS (caráter, formação, produção, natureza) com o sufixo advindo do latim, CULTURA (ato de tratar, cuidar, desenvolver). (ORIGEM DA PALAVRA, 2018)

O culto ao corpo não é exclusividade dos tempos modernos, vemos referências a corpos musculosos e definidos desde a Grécia antiga em gravuras e estátuas de deuses e semideuses como as de Héracles de Farnese (230 d.C.) e Laocoon ou Apolo (século I) (Bittencourt et al apud Neves, 2012).

Figura 1: Estátua de Hércules de Farnese (esquerda) e Laocoon (direita)

Segundo Leighton apud Neves (2012) a mitologia grega confere o início do exercício de resistência de peso progressivo ao conto de Milo, que viveu em Crotona na Grécia. De acordo com a história, Milo queria se tornar o homem mais forte não apenas de toda a Grécia, mas de todo mundo, para conseguir esta meta, ele começou levantando um bezerro quando ainda era jovem, à medida que o bezerro crescia, Milo aumentava sua força pois todo dia o erguia. Quando o touro atingiu a idade adulta, Milo havia desenvolvido força suficiente não só para levantar o animal, como também para carregá-lo nos ombros.

Na cidade de Olímpia foram encontradas em escavações pedras datadas do século VI a. C. com formato semelhante aos halteres como usados hoje nas academias (NEVES, 2012). Na Idade Média, segundo Figueiredo apud Neves (2012), existiam barras de Ferro com duas grandes bolas nas extremidades. No século XVIII surge o Tratado de Educação Física e Moral, escrito pelo espanhol Francisco Amarós, que preconizava a utilização de halteres para melhorar o físico (TUBINO, 2007).

Após um longo período histórico de anulação do corpo em detrimento de valores religiosos vemos a retomada do interesse pelo corpo no período denominado Renascença. Segundo Lichtenstein apud Daré (2015) o século XV inventou o corpo, estudando anatomia, formas e expressões.

Vemos o início da busca pelo utópico corpo perfeito. É neste período que Leonardo da Vinci revisita a obra de Marco Vitruvius Pollio, onde há uma definição métrica do que seria o corpo perfeitamente simétrico em uma parte de sua obra intitulada *De Architectura*, e representa-o de forma gráfica (IBIDEM).

Figura 2: Homem Vitruviano, desenhado por Leonardo Da Vinci

Mas é no séc. XIX que vemos os primeiros relatos de desenvolvimento do corpo para fins estéticos e competições de culto ao corpo.

2.1.1 Da subcultura às Olimpíadas

Os atletas profissionais são, em muitos aspectos, o homem sagrado da nossa cultura: eles se entregam a uma busca, suportam grandes privações e dores para se realizarem nela, e desfrutam de relacionamentos com a perfeição que admiramos, recompensamos e adoramos ver, mesmo que não tenhamos desejo de perfazer esta trilha por nós mesmos. Em outras

palavras, eles fazem isso “por nós”, se sacrificam pela nossa (imaginamos) redenção. (WALLACE, 2009, tradução nossa)²

No final do século XIX a tradição antiga de levantar pedras evoluiu dentro do esporte moderno de levantamento de peso. À medida que o esporte se desenvolveu, enfrentou diferentes aspectos em diferentes culturas.

Na Europa o levantamento de peso era uma forma de entretenimento da qual alguns atletas começaram a tirar seu sustento. A aparência física nesta modalidade de entretenimento não importava muito.

Na América do Norte surge um interesse e preocupação por cuidados com o corpo e saúde. A mudança populacional dos campos para cidade, o aumento da quantidade de alimentos processados e a invenção do automóvel traziam as comodidades da vida moderna, porém também arrastava seus problemas, a vida se tornara extremamente sedentária, problemas de saúde ligados a uma nova cultura de alimentação e condições de stress cada vez mais constante (SCHWAZENEGGER, 1998).

Na virada do século XX um homem se destaca fazendo shows e apresentações públicas de seu corpo e sua força, Eugen Sandow se torna uma celebridade da cultura física. Sandow, além de ficar conhecido por suas apresentações de demonstrações de força ao levantar grandes cargas, ficou também conhecido pelo corpo atlético diferenciado da época. Ao mostrar seu corpo seminu em apresentações ao ar livre Sandow chocou a sociedade vitoriana que não estava acostumada com o nu masculino. (IBIDEM, p. 32)

Após ganhar o concurso *World's Strongest Man*, em 1889, Eugene Sandow iniciou uma turnê que abrangeu partes da Inglaterra, Austrália, África do Sul e a Europa Continental. Nesta turnê, além das demonstrações habituais de força como a torção de ferros, lutas com leões, e levantamento de pesos, Sandow decidiu adicionar em suas *performances* um novo elemento: a apresentação de poses acompanhada de uma apresentação musical. Dutton (1989) relata que Sandow introduziu na época uma

² “Professional athletes are in many ways our culture’s holy man: They give themselves over to a pursuit, endure great privation and pain to actualize themselves at it, and enjoy relationships to perfection that we admire, reward and love to watch, even though we have no desire to walk that road ourselves. In other words, they do it “for” us, sacrifice themselves for our (we imagine) redemption.” (WALLACE, David Foster. 2009)

performance com acompanhamento musical dentro de uma caixa de vidro, realizando uma série de poses que evidenciavam sua musculatura. Este é o primeiro relato de intersecção entre música e culturismo que pudemos averiguar.

Tendo tino empreendedor, Sandow soube aproveitar de suas características físicas para lucrar dando início à indústria do *fitness*³. Começou a dar palestras, criar concursos e escrever revistas e livros com técnicas e lições para quem quisesse ter a estética dele, assim como houve também aumento no interesse por aparelhos de ginástica e pesos na época.

Figura 3: Eugen Sandow em uma apresentação no início do séc. XX

Foi no primeiro quartel do séc. XX que começou a se pensar no uso do treinamento de esforço resistido como forma de moldar o corpo e não somente para aumento de força (SCHWARZENEGGER, 1998).

Após a Segunda Guerra Mundial surgem a companhia Weider Barbell, indústria de equipamentos, e a revista *Muscle Builder* (LEIGHTON apud NEVES, 2012).

³ Insere-se dentro deste contexto os segmentos comerciais de roupas para academia, pesos, suplementos, revistas especializadas entre outros.

Na década de 60 o esporte ganha amplo espaço na América do Norte, mais específico na Califórnia, é nesse período que surgem nomes expressivos no esporte como Arnold Schwarzenegger, Frank Zane e Sergio Oliva (NEVES, 2012).

A crescente indústria Hollywoodiana, cravada no coração da Califórnia, ajuda a fomentar, ainda que discretamente em suas primeiras décadas, o culto ao corpo. Séries de filmes como Tarzan, se apoiaram em atletas para protagonizarem seus personagens. Podemos citar Johnny Weissmuller e Buster Crabbe, ambos nadadores profissionais que emprestaram seus físicos para interpretar o rei da selva.

No início dos anos 80 os corpos musculosos e definidos chegam de vez a Hollywood, filmes como “Conan, o Bárbaro” (1982), “As aventuras de Hércules” (1985) estreiam com fisiculturistas como protagonistas. Uma onda de filmes de ação toma o cinema e atores como Sylvester Stallone (Rambo, Stallone: Cobra, O Juiz), Arnold Schwarzenegger (Exterminador do Futuro, Comando para Matar, O predador) e Jean Claude Van Damme (Leão Branco, Cyborg, O Alvo) começam a moldar uma tendência nova nos conceitos da estética popular.

A cultura do corpo propagada pelos filmes é um efeito da mídia capaz de perpetuar estereótipos com a mesma eficácia com que ajuda a eliminá-los. (STEINBERG apud JORDÃO, 2013).

A partir da década de 80 vemos um esporte com parâmetros bem definidos, com federações internacionais fazendo campeonatos em todos os cantos do mundo, a cultura fitness se populariza e gradativamente ganha mais adeptos ao decorrer dos anos criando a necessidade de discutirmos e analisarmos sobre diferentes pontos de vista a capacidade de comunicação que existe dentro dos corpos construídos pelo esporte.

Em 2016 foi anunciado a participação das categorias *classic physique* e *woman's bikini* da federação IFBB, nos jogos pan-americanos de Lima em 2019 (IFBB, 2018) e consequentemente nas olimpíadas de Tokyo (KARONGA, 2016) em 2020.

2.1.2 Em busca do corpo perfeito: Competições de Fisiculturismo

Como o próprio nome sugere, fisiculturismo é o culto ao corpo, a busca por um físico perfeito. Neste esporte, não é medida a força, o condicionamento físico ou a resistência, mas apenas a beleza estética do corpo. Trata-se de uma modalidade esportiva relacionada a musculação, mas no sentido estético e artístico da forma física, e não

da carga ou peso levantado. Ganha o físico que melhor representar uma estátua grega. Os critérios de julgamento do *Bodybuilding* (Fisiculturismo) são: Volume Muscular, Definição Muscular, Simetria (se existe diferença entre os músculos de ambos os lados do corpo) e Proporção (tamanho e harmonia de cada membro e grupo muscular em relação ao todo). Todos esses quesitos possuem o mesmo grau de importância. O árbitro avalia o melhor conjunto. (PIERIN, 2016)

Há no mundo, hoje, muitas federações mundiais com programação internacionais, com eventos regulares acontecendo quase que todo final de semana em algum lugar do mundo.

Podemos citar algumas como: *British Natural Bodybuilding Federation* (BNBF); *International Federation of Body Builder* (IFBB); *Miami Pro Events*, *National Amateur Bodybuilders Association* (NABBA); *Northern Ireland Fitness Model Association* (NIFMA); *Physical Culture Association* (PCA); *The Natural Physique Association*; *United Kingdom Bodybuilding and Fitness Federation* (UKBFF); *World Beauty Fitness e Fashion* (WBFF); *National Physique Committee* (NPC); *World Bodybuilding Federation* (WBF); *World Natural Bodybuilding Federation* (WNBF)

Cada federação é responsável pela criação de seu regulamento, onde especificam os critérios de avaliação, vestimentas permitidas nas competições, tipos de testes *antidoping* entre outros parâmetros.

Para esta pesquisa optamos por adotar as regras da Federação Internacional de Bodybuilding (IFBB).

A IFBB foi fundada pelos irmãos Joe e Bem Weider em 1946 na cidade de Montreal, Canada. Após 72 anos de existência a federação está presente em mais de 191 países.

Há ao todo 10 categorias que podem ser disputadas pela IFBB. São elas: *Men's Bodybuilding*; *Men's Classic Bodybuilding*; *Men's Physique*; *Mens Fitness*; *Men's Fit Model*; *Women's Fitness*; *Women's Bodyfitness*; *Women's Bikini Fitness* e *Women's Physique*. (SANTOJA, 2016)

Cada categoria tem parâmetros diferentes de comparação e buscam premiar diferentes tipos de físicos. Como na implementação de nosso sistema interativo iremos utilizar as poses da categoria *Men's bodybuilding* (fisiculturismo masculino) iremos nos ater a suas especificidades.

Na categoria fisiculturismo masculino os parâmetros analisados são: volume muscular, simetria entre as seções do corpo e baixo percentual de gordura corporal. O indivíduo que conseguir apresentar o maior volume muscular com definição dentro de um corpo harmonioso e simétrico será premiado. Os competidores são divididos por peso havendo uma categoria de 5 kg em 5 kg de 60 kg até acima de 100kg.

Dentro das competições ocorrem dois confrontos iniciais. No primeiro confronto o atleta é colocado ao lado de mais 5 competidores para execução das 8 poses obrigatórias, são elas: expansão de dorsal de frente, duplo bíceps de frente, peitoral melhor lado, tríceps melhor lado, expansão de dorsal de costas, duplo bíceps de costa e mais musculosa⁴.

Após o primeiro confronto ocorre a escolha dos 10 finalistas, esses voltam individualmente ao palco para realizar sua rotina de poses coreográficas. Neste momento o atleta dispõe de 1 minuto para realizar uma rotina livre de poses junto a uma trilha musical escolhida previamente. Nesta etapa o atleta é avaliado pela criatividade, ritmo e sincronia das poses em relação a música.

Foi baseado nesta parte da competição que o atleta deste esporte foi escolhido para ser o ator do sistema interativo, servindo como um controlador gestual.

2.2 Instrumentos Musicais Digitais (IMDs)

Um instrumento que usa geração de som através de computação pode ser chamado de instrumento musical digital (IMD) e consiste em superfície de controle ou controle gestual, o qual pode controlar parâmetros de sons sintetizados⁵ (MIRANDA & WANDERLEY, P.1,2006, tradução nossa.)

Com a introdução de computadores comerciais de baixo custo e alto poder de processamento tornou-se comum a utilização destas máquinas para processamento

⁴ Todas estas poses estão referenciadas no anexo B presente ao fim desta dissertação.

⁵ “An instrument that uses computer-generated sound can be called a digital musical instrument (DMI) and consists of a control surface or gestural controller, which drives the musical parameters of a sound synthesizer” (MIRANDA & WANDERLEY, P.1, 2006)

musical em tempo real. Hoje em dia um computador simples tem capacidade para trabalhar com processamento de áudio com uma qualidade equivalente ao CD (taxa de amostragem de 44.100 Hz e 16 bits de resolução digital). Neste contexto, observamos tornar-se comum a criação de instrumentos musicais digitais e uma ampliação no interesse por formas de controle alternativas aos controladores usuais, como é o caso dos controladores gestuais.

A evolução tecnológica na área musical, principalmente no desenvolvimento de instrumentos digitais, está atrelada às necessidades do meio de produção musical de cada período da história. Interfaces que proporcionem: facilidade de uso; rapidez na compreensão de seu funcionamento; capacidade de readaptação/reconfiguração; e transparência parecem ser adequadas ao momento em que vivemos. (IAZZETTA, 1998)

Segundo Patrício apud Wanderley (2010) o modelo básico de instrumento digital se dá a partir de seus componentes/estágios: *interface* gestual (ou dispositivo de entrada) e unidade de geração sonora. A interface gestual corresponde ao decodificador de gestos físicos realizados pelo *performer* em informação binária, utilizada para controlar parâmetros da unidade de geração sonora. Porém para que a relação seja efetiva precisamos definir a estratégia de mapeamento. Segundo Hunt et al (2000) é a correspondência entre os parâmetros de controle e os parâmetros de síntese sonora que definem um mapeamento.

Neste trabalho optamos por focar nos IMDs, catalogados por Miranda e Wanderley (2006), na classe de controladores gestuais alternativos, ou seja, aqueles que não derivam de um instrumento acústico. Perez apud Mulder (2016) classificará sensores de movimento, como o Kinect, na categoria de controles imersivos, sendo aqueles que disponham de qualquer tipo de contato do usuário para que haja controle.

Miranda e Wanderley (2006, p.12) dividem as formas de captura de gestos físicos em: captação direta, captação indireta e captação fisiológica. A captação direta é aquela que os sensores captam diretamente o movimento do corpo. Captação indireta pressupõe a necessidade de aquisição de dados do *performer* através de um intermediador, como um microfone. E o último modo diz respeito a mapeamentos que utilizem sensores que captam estímulos fisiológicos como eletromiograma e eletroencefalograma.

Após a leitura do sensor, inicia-se o mapeamento das mensagens para o gerador de som. Este mapeamento pode ser definido através de como se dá a correspondência

entre gesto e som. Miranda e Wanderley (2006, p. 14) irão classificar em três modos de interação básicos: um-para-um (*one-to-one*), um-para-muitos (*one-to-many*) e muitos-para-um (*many-to-one*).

O primeiro modo (*one-to-one*) significa que um parâmetro de controle influenciará um parâmetro de síntese sonora. O segundo modo (*one-to-many*) um parâmetro de controle controla mais de um parâmetro de síntese sonora. Por fim o terceiro modo (*many-to-one*) vários parâmetros de controle podem ser usados para determinar apenas um parâmetro de síntese.

2.2.2. Sistemas gerativos de Música através de Interfaces de Usuário Naturais

Dentre os sistemas gerativos de música em tempo real, neste trabalho aborda-se o caso das Interfaces de Usuário Naturais (ou NUI – *Natural User Interface*, como abordado em CARDOSO, 2013). Nesta seção são apresentados alguns trabalhos de referência na área que fazem uso de NUIs, como o equipamento Kinect utilizando as formas de mapeamento *one-to-many* e *one-to-one*, descritas acima.

O primeiro foi escrito por Matthias Kronlachner e publicado nos anais do congresso LIPAM (*Live Interfaces: Performance, Art, Music*) em 2012 e descreve a criação de um ambiente usando o Kinect para manipular a síntese de áudio em tempo real durante o desempenho de um dançarino. O segundo artigo, de 2015, relaciona o esforço de dois pesquisadores na criação de um sistema interativo que possa ser usado satisfatoriamente por artistas experientes e pessoas com algum grau de deficiência. Através da junção de uma câmera Ethernet e do Kinect os autores criaram um sistema preciso de rastreamento de movimento (BERGLAND et al, 2015). O terceiro artigo, traz uma pesquisa brasileira feita na Universidade de Campinas apresentada na Conferência Internacional IEEE Systems, *Man and Cybernetics*, mostrando o processo de criação de uma peça interativa que usa quatro tipos diferentes de sensores. O quarto artigo traz o trabalho de Alexander Jensenius (2013) que utiliza *motiongram* como informação para controlar síntese sonora. Por último é apresentado o sistema *Crossole*, um sistema musical interativo que trabalha

com blocos de acordes criando estruturas musicais complexas, sendo que um gesto captado pode alterar acordes e o modo de sequenciamento de suas respectivas notas.

Após a descrição de cada artigo, consideramos alguns pontos de convergência, observando aspectos técnicos e circunstanciais nos estudos de caso.

Vertimas - übersetzen é uma peça criada por Matthias Kronlachner do Instituto de Música Eletrônica e Acústica da Universidade de Música e Artes Performáticas de Graz. Foi criada para dançarino, som e projeção tendo o Kinect como o tradutor dos movimentos do corpo para o som, transformando o corpo do dançarino em um hiper-instrumento. As informações de profundidade captadas são usadas para enviar o contorno do dançarino e projetar informações visuais em seu corpo em tempo real.

Figura 4: Estrutura de hardware e software da peça Vertimas - übersetzen (KRONLACHNER, 2012)

As bibliotecas de filtragem e análise de fluxo de dados foram desenvolvidas em duas instâncias separadas usando o software Pure Data (Pd), uma para trabalhar com informações visuais e outra, para informações sonoras, traduzindo dados de rastreamento em dados de controle separados para conteúdo sonoro e visual, afastando assim eventuais problemas de processamento.

A primeira instância foi designada para renderização visual e mapeamento do esqueleto, e a segunda para áudio e controles recebidos via OSC para síntese sonora.

A primeira apresentação da Vertimas überstzen foi no Jauna muzika festival 2012 em Vilnius, Lituânia, apoiada pela Academia Lituana de Música e Teatro.

Figura 5: Diagrama de interação MotionComposer (BERGLAND et al, 2015)

A primeira área de aplicação pensada para o Motion Composer (MC) foi voltada para pessoas com deficiência (BERGLAND et al, 2015). Trata-se de um dispositivo que usa o monitoramento de movimento baseado em sensor de vídeo e 3D para transformar os movimentos captados em música.

O hardware MC combina dois sensores de rastreamento de movimento: um Kinect e uma câmera Ethernet de 1,3 megapixels. A câmera fornece uma imagem de baixa latência e alta resolução, enquanto o sensor Kinect 3D é usado para rastrear a localização e a postura do usuário.

As informações de ambas as câmeras são interpoladas por um *software* desenvolvido por Simeno Ghisio e Paolo Coletta no ambiente *EyesWeb*⁶, produzindo mais de vinte *streams* de controle que são enviados através de OSC para o *software* de síntese musical (CSound, PureData e SuperCollider) em tempo real. Os principais parâmetros relevantes utilizados são: quantidade de atividade, centro da posição horizontal do usuário e altura.

O software foi criado com base em três princípios: O primeiro foi a inclusão, para que possa ser usado igualmente por pessoas com e sem deficiência. O segundo de ser

⁶ Plataforma livre que dá suporte à criação e desenvolvimento de sistemas e interfaces multimodais em tempo real.

sinestésico, com o objetivo de ser um instrumento musical e um dispositivo de dança. Por último, o sistema precisa ser artisticamente satisfatório e divertido.

Considerando a inclusão como principal preocupação dos desenvolvedores, o sistema possui três modos de operação: sala, cadeira e cama, fazendo referência a capacidade de mobilidade do usuário, adequando-se o controle também para cadeirantes e acamados.

O MC tem seis tipos diferentes de ambientes virtuais e as estratégias de mapeamento foram diferentes para cada ambiente.

Na peça *Jeu de Modes*, o ambiente *particles* foi escolhido pelos intérpretes. O ambiente é baseado em objetos de som curtos com duração inferior a um segundo, como remete seu nome.

A ideia de mapeamento em *particles* baseia-se em: A localização do usuário ortogonal em relação a câmera dentro da sala é o que determinará a escolha das partículas de som. A quantidade de atividade, que determinará a frequência com que a(s) partícula(s) escolhida(s) serão acionadas e a estratégia de mapeamento dos parâmetros de desempenho e controle é uma combinação de *one-to-one*, *one-to-many* e *many-to-one* (MULDER, 2000).

O mapeamento citado foi usado na peça *Jeu de modos* pelo grupo Palindrome, por exemplo. Nesta parte, o parâmetro de quantidade de atividade foi usado para controlar o ganho de saída de sete fluxos rítmicos sincronizados, que contém cadeias de amostras de som. Nesta peça, é especificamente a quantidade de atividade de todos os três dançarinos juntos que são rastreados.

O MotionComposer é uma ferramenta complexa com várias possibilidades de programação. Ao combinar seis ambientes interativos com três modos de controle, cria-se a possibilidade de que qualquer movimento possa ser transformado em som.

Os ambientes podem exigir movimentos criativos na maioria dos usuários, incentivando e motivando a superação de desafios sendo uma ferramenta no tratamento de deficiências mentais ou físicas.

MinDSounDS é uma peça interativa multimodal para computador, movimento, WiiMote, flauta, Brain-Computer Interface (BCI) e microfone, que teve sua estreia na conferência Generative Arts 2014 (TAVARES et al, 2015).

A peça usou quatro tipos de sensores, como sensores de movimento (Kinect), análise de áudio em tempo real com técnicas de Recuperação de Informações de Música (MIR), WiiMote (acelerômetro) e Epoc (BCI) para alcançar a finalidade dos compositores. O projeto foi desenvolvido por uma equipe de pesquisa do Núcleo Interdisciplinar de Comunicação de Som (NICS) da Universidade de Campinas, Brasil, e ilustra como experiências estéticas podem ser relacionadas a uma organização emergente da interação entre os artistas e um ambiente virtual, aumentando as possibilidades expressivas dos dançarinos. A peça narra a história de um avatar virtual, uma projeção humanoide em tela, que aprende movimentos de um dançarino humano e desenvolve seus próprios movimentos. Todo o processo é mediado por artistas humanos que interagem entre si e com o ambiente virtual.

Na peça, os sensores foram usados para gerar projeções visuais e som, cada qual dentro de um design de contexto especificado pelos autores, considerando as limitações e a possível aplicação artística. Um dos sensores que foi usado na peça foi o Kinect, por ser um sensor de movimento capaz de ler as posições dos membros e interpretá-los relacionado ao torso do artista, transformando a kinesfera⁷ em dados manipuláveis.

O uso do Kinect permitiu mapear os movimentos de mãos em torno da kinesfera para controlar a espacialização, a seleção de amostras e o controle de vídeo. A distância entre as mãos foi mapeada para mudar o tom e a seleção da amostra musical. A velocidade dos pés controla a intensidade do som e a posição relativa dos pés controla o nível de granulação. Nem todos esses mapeamentos são usados ao mesmo tempo, mas dispersos em movimentos particulares da peça.

A escolha do WiiMote foi devido ao seu acelerômetro de três eixos. A utilização do dispositivo permite utilizar dados do acelerômetro, bem como os botões do hardware para controlar disparos de informações sonoras.

⁷ A noção de kinesfera é apresentada por Rudolf Laban que a define como "a esfera periférica em volta ao corpo que pode ser alcançada estendendo os membros sem que se tire o pé de suporte do local inicial." (LABAN e ULMANN, 1966)

Figura 6: Fluxograma do sistema interativo MindSounDS (TAVARES et al, 2015)

Comparando com o Kinect, o WiiMote exibe uma resposta mais rápida, mas também produz significativamente mais ruído na informação (IDEM). O dispositivo foi usado para controlar uma percussão virtual e o mapeamento de entrada do dispositivo foi feito pressionando-se o botão A de controle.

Um movimento imitando um gesto de tapa é usado para tocar a percussão. Os botões direcionais controlam os dados de gravação para o filtro adaptável. O acelerômetro foi usado para alternar a interpolação do filtro e o botão B para ligar o filtro. O gesto de tapa foi feito detectando valores de aceleração em qualquer eixo em limiares pré-definidos.

O Epoc é um dispositivo comercial que fornece uma BCI (*Brain-Computer Interface*). O sistema consiste em muitos eletrodos e um acelerômetro que pode fornecer uma leitura de eletroencefalograma e a possibilidade de mapear pensamentos para controle de *software*. Devido a problemas de instabilidade e a quantidade de falsos positivos e falsos negativos gerados, o uso deste BCI foi restringido apenas para o controle de parâmetros de alto nível. No contexto musical, o único uso de BCI na peça remete a um movimento onde o avatar aprendia os movimentos do dançarino, sendo usado para desencadear um próximo passo no processo de aprendizagem, correspondente a um novo valor máximo para o tempo entre um passo e outro. O próximo intervalo de

tempo máximo é definido com os valores anteriores menos uma fração do tempo decorrido desde o tempo do passo anterior. Cada passo é marcado por um som de campainha. Isso possibilitou estimar a duração do movimento que foi útil para a interação com outros músicos.

A recuperação de informações musicais (MIR) foi o último processo de manipulação sonora utilizado na peça. Adquirindo um som de flauta acústica através de um microfone, foi possível usar características espectrais e temporais para modificar partes visuais das peças.

Dois recursos de áudio foram usados para controlar valores contínuos no processamento de vídeo, com o recurso de croma determinando um intervalo de tonalidades, enquanto a intensidade determina a luminosidade das texturas renderizadas em vídeo. Isto é, as notas se transformam em cores projetadas, um processo que foi feito aleatoriamente.

Usando essa combinação de sensores e métodos de mapeamento, o sistema foi capaz de superar a deficiência de cada sensor individualmente e usar suas melhores características de uma forma artística satisfatória.

Kinectofon: *Perfoming with Shapes in Planes*, é um instrumento digital projetado para criar sons através de movimentos das mãos em um ambiente 3D (JENSENIUS, 2013). Utilizando os dados recebidos através das câmeras RGB e infravermelho do Kinect cria-se a fonte do material de controle para o processo de sonificação que utiliza como base as técnicas de síntese da inversão da transformada de Fourier.

A implementação foi feita utilizando-se os *softwares* Max/MSP/Jitter e componentes do *open framework* Jamoma.

A técnica de sonificação, segundo o autor, é denominada *sonomotiongram*, palavra derivada de *motiongram* cujo conceito é a exibição do movimento espaço-temporal a partir de uma gravação de vídeo onde, a partir do processamento deste, faz-se a média sobre cada linha na matriz de um *frame* e qualquer alteração no eixo vertical cria uma informação de comando que é utilizado para controle de disparos sonoros.

Ao captar *frames* através do Kinect do ambiente em tempo real cria-se a possibilidade de acrescentar filtros no quadro captado para visualizar a silhueta ou o primeiro plano dentre outros. Ao todo há sete possibilidades de imagens que podem ser enviadas ao módulo do *motiongram* para serem processadas posteriormente.

Figura 7: Diagrama de processo do ambiente Kinectofon (JENSENIUS, 2013)

A técnica de *sonomotiongram* baseia-se em tratar o *motiongram* captado da imagem como um espectrograma de áudio. Apesar do espectrograma de áudio representar as diferenças de energia de bandas de frequência enquanto um *motiongram* consiste na redução da exibição de uma série de imagens em movimento, ambos compartilham uma propriedade: o desdobramento temporal de formas. O eixo Y em um *motiongram* representa movimento vertical, o qual pode ser associado a alternância de *pitch*, deixando o controle do sistema intuitivo e fácil de aprender.

A técnica normal de *motiongram* baseia-se na média de cada linha da matriz da imagem, o que significa que qualquer movimento nos eixos leva a uma alteração visual e, portanto, sonora. Com a capacidade do Kinect em filtrar aspectos da imagem cria-se a possibilidade de combinar vários *motiongrams* em tempo real culminando em possibilidades de uma maior criação sonora.

A câmera de profundidade do Kinect permite a criação de uma “caixa virtual” que possibilita a performance. O artista pode mover as mãos dentro e fora da caixa controlando *pitch*, timbre, início e fim como se tivesse o controle tátil do som.

Nosso último sistema a ser analisado é o Crossole, um meta-instrumento onde os elementos musicais são visualizados como um conjunto de blocos virtuais que remetem a estrutura de um jogo de palavras cruzadas que podem ser interligados através de

controle gestual mediado pelo sensor Kinect, criando material sonoro em diferentes profundidades e modos. Em síntese o instrumento consiste em uma interface gestual para composição, improvisação e *performance* usando Kinect.

Com a implementação de controle caracterizada como *many-to-one* (MIRANDA; WANDERLEY, 2006) entre o gesto do músico e o som produzido é possível controlar acordes e estruturas complexas com apenas um gesto ao invés de tocar nota por nota.

O conceito de operação no Crossole se divide em três formas de visualização. O primeiro é o *crossword view*, que consiste no ambiente em que é montado as estruturas de acordes que serão tocadas. Basicamente o músico escolhe os acordes que deseja que sejam tocados alinhando-os em qualquer eixo (x, y) no plano e depois configurando como será a sequência de leitura dos acordes. Após configurado, o sistema toca em *loop* a sequência até que seja alterada alguma função pelo músico.

O segundo ambiente de controle é denominado *grid view*. Neste ambiente o músico pode remanejar a forma de arpejo dentro de cada acorde antes pré-determinado. Essa função permite que o músico possa trabalhar configurando controles de baixo nível, podendo escolher nota por nota.

O terceiro ambiente é denominado *timbral view*. Neste modo o músico pode criar *soundscapes* aplicando vários efeitos como filtros, *phasers* e *delays* através de gestos livres das mãos. Enquanto está neste modo de controle o músico pode alternar para o modo *grid view* e alterar o *pitch* em que os efeitos estão atuando.

O controle gestual de Crossole baseia-se em quatro operações maiores: alocação de blocos de acordes, controle do cursor, desenho do *pitch* e a manipulação do timbre.

O sistema foi implementado com a utilização do Kinect para a captura de gestos. As mensagens OSC são recebidas pelo framework programado em C++ que mapeiam as coordenadas tridimensionais em variáveis de controle que se alteram de acordo com o ambiente que o músico está. As variáveis de controle são concatenadas em OSC e enviadas para o *software* MAX/MSP. Este por sua vez processa as variáveis em ações musicais e envia uma mensagem para o *software* Ableton Live realizar a síntese sonora.

Figura 8: Fluxograma do sistema interativo Crosssole (SENTURK et al, 2012)

O sistema interativo Crosssole foi *performedo* no Listening Machines 2011, na série de concertos desenvolvida pelo Georgia Tech Center for Music Technology.

Após a análise de todos os sistemas interativos, o primeiro ponto a destacar é a semelhança no *design* de todos os projetos. Criar ambientes onde não-músicos possam controlar e criar materiais sonoros revela o interesse constante da pesquisa interdisciplinar para expandir as áreas tocadas pela música através de interfaces de controle de fácil aprendizado, quando o Kinect se torna uma ferramenta vantajosa visto que o intérprete faz uso de seu próprio corpo para controlar o sistema.

O Segundo ponto, ainda em atenção ao usuário, remete à preocupação para que a forma de mapeamento entre gesto e som seja a mais intuitiva possível. Esta preocupação mostra o esforço dos projetistas em adequar o sistema da melhor maneira para que a curva de aprendizado do sistema seja a menor possível.

A fusão entre as práticas artísticas de uma maneira intrínseca é clara nos artigos. Através da expressão artística já conhecidas de várias vertentes como por exemplo dançarinos, como no caso dos artigos estudados, é possível controlar parâmetros musicais, abrindo novas possibilidades para que artistas possam se expressar e proporcionar novas experiências para o público.

Sobre a relevância das áreas de entrelaçamento, vemos no segundo artigo por exemplo, a função inclusiva que os sistemas musicais interativos podem ter ao se pensar sobre seu uso para pessoas com deficiência e como eles podem auxiliar na reabilitação

física e emocional destas, mostrando a importância do diálogo entre diferentes áreas para beneficiar um maior número de potenciais usuários.

A abordagem de cada sistema foi semelhante em aspectos gerais, mas diferente em pontos particulares. Todos os sistemas abrangidos utilizaram o sensor de movimento Kinect em alguma instância para a captura de movimento, evidenciando sua utilidade para fins artísticos em meio a sensores na pesquisa atual.

Abaixo comparamos as diferenças e semelhanças na montagem dos sistemas e do tipo de mapeamento entre gesto/som predominante em cada projeto.

Tabela 1 - Comparação de aspectos técnicos dos projetos

Sistemas	Sensores	Middleware	Software Musical	Mapeamento
Vertimas	Kinect	Pure Data	Pure Data	one-to-one
Motioncomposer	Kinect Camera Ethernet	Open Framework	Super Collider Pure Data Csound	one-to-one/ one-to-many/ many-to-one
Mindsounds	Kinect BCI WiiMote Microfone	-	-	one-to-one
Kinectofon	Kinect	Open Framework	Max/MSP Jitter	one-to-one
Crossole	Kinect	Max/MSP	Ableton Live	many-to-one

Pelos estudos de caso apresentados, analisando-se os aspectos técnicos dos sistemas e os resultados alcançados pelos trabalhos, podemos concluir que o sensor Kinect demonstra eficácia na captação de gestos que utilizam movimentos maiores, sendo que nos dois trabalhos que buscaram pela captação de movimentos de menor amplitude foram utilizados ainda outros sensores complementares.

Pode-se observar também que a maioria dos estudos optou pelo mapeamento relacionando um gesto a um elemento musical, o que torna o controle musical mais simples embora também limite as possibilidades de disparo do intérprete

Os *softwares* mais utilizados são os denominados *open source*, tendo em vista que o desenvolvimento das ferramentas que explorem as funcionalidades do Kinect nestes ambientes é constantemente alimentado por comunidades virtuais de programadores independentes.

O principal ponto que pode ser visto através dos trabalhos analisado é como o uso de interfaces gestuais possibilita que a música transpasse diversas áreas, tocando públicos distintos e possibilitando não-músicos a trabalhar parâmetros musicais com pouco tempo de aprendizado, tarefa difícil em instrumentos tradicionais.

Embora nossa abordagem nesta etapa tenha excluído quantificar e comparar a facilidade de uso e grau de captura de expressividade existente entre os casos, todos os sistemas analisados foram bem sucedidos em apresentações artísticas para o público, e relataram os aspectos funcionais no uso do sensor Kinect como uma superfície de controle IMD efetiva para trabalhos interativos, fornecendo suporte e fundamento para projetos de novas gerações de mapeamento de sistemas de música interativos que almejam tanger diferentes áreas simultaneamente

3. Projeto de um sistema interativo musical controlado por performance coreográfica utilizando o *Kinect*

Neste capítulo apresenta-se o *Body Building Music*, o sistema de criação musical concebido nesta pesquisa, a partir das motivações colocadas e norteado pelos sistemas interativos de referência estudados, montamos uma estrutura que permite a um atleta controlar a geração de uma peça musical em tempo real.

A raiz da ideia de implementação do sistema nasceu após 4 anos de vivência em meio ao esporte e dentre campeonatos de fisiculturismo. Ao experienciar todos os processos que o esporte exige, dos treinos a meticulosa escolha de cada pose de sua apresentação, percebe-se que as preocupações do atleta de fisiculturismo podem ser comparadas às de duas outras categorias artísticas.

Primeiro podemos traçar um paralelo entre as preocupações do fisiculturista e um escultor renascentista. Guardada as diferentes ferramentas e materiais de ambos, podemos dizer que a busca pelas proporções e simetria ideais do corpo humano são o foco de trabalho tanto do fisiculturista moderno, quanto dos corpos esculpidos com proporções matemáticas por grandes escultores no século XV.

A segunda tangente se dá com a dança. Não é raro observar que fisiculturistas, profissionais e amadores, colocam em meio a sua rotina de treino, aulas de dança. Arnold Schwarzenegger, por exemplo, tomou aulas de ballet na turma de dança da UCLA em 1969 afim de melhorar sua rotina de poses. (SCHWARZENEGGER, pg. 100, 1977)

Dentre a linha tênue entre esporte e arte em que a performance se equilibra, podemos poetizar lances esportivos e comparar obras de arte sob crivos competitivos sem maiores embates. Sendo assim o caminho contrário, tornar tal linha seccional ainda mais turva, encontra seu propósito ao tornar o atleta um pouco mais artista e situar o compositor no meio esportivo, dando a possibilidade de ambos pensarem e trabalharem funções fora de suas zonas de atuação habitual.

Para a implementação de um sistema que se propões dialogar entre dois universos, a saber: música e fisiculturismo, fez-se necessário transpassar por uma terceira área, a programação computacional.

O sistema propõe disponibilizar ao atleta três níveis de controle sobre a peça musical:

- 1) Primeiro: nível de controle interativo da estrutura da peça musical.
- 2) Segundo: nível de controle de efeitos.
- 3) Terceiro: nível de controle de dinâmica.

O primeiro nível possibilita ao atleta, através da realização das sete poses compulsórias⁸, disparar ou pausar blocos instrumentais da orquestração. Podemos fazer uma comparação desse controle a ação de um regente perante a orquestra. Sinalizando para as famílias instrumentais o momento de tocar e o momento de se manter em silêncio.

Podemos comparar os outros dois níveis com a atuação de um engenheiro de som em tempo real, onde se controla parâmetros do âmbito das pós-produção do áudio.

No segundo nível, responsável pela adição e controle de parâmetro dos mesmos, o atleta, através do movimento contínuo de suas duas mãos espelhadas no eixo horizontal, pode alterar a frequência central de um filtro semi-paramétrico⁹, que age sobre as trilhas que estão ligadas neste momento.

O terceiro nível fica responsável pelo controle do volume, isto é, por determinar os valores dinâmicos da trilha musical ao longo da peça. Através do movimento das mãos, paralelas entre si e se movendo no eixo vertical, o atleta pode aumentar ou diminuir o volume geral das trilhas que estão sendo tocadas.

3.1 Concepção e desenvolvimento do sistema

A criação do sistema foi realizada em três etapas fundamentais. A primeira (planejamento) foi responsável pela definição do mapeamento entre gestos e som, aspectos de controle, como a forma de ativação e a escolha dos *softwares* e *hardwares* para compor o sistema que melhor se enquadravam na proposta. Na segunda etapa (desenvolvimento e integração) foi realizada a montagem dos componentes do sistema, blocos de software (programação de software), e a integração destes componentes. Por fim, na terceira etapa se deu a implementação e teste do processo criativo da composição interativa.

⁸ Para maiores informações sobre as regras e poses das competições de fisiculturismo ler o capítulo 2.1.2.

⁹ Os equalizadores semi-paramétricos diferenciam-se dos paramétricos por terem seu Q (fator de qualidade = $f_c/\text{largura de banda}$) fixo.

O cerne do projeto encontra-se no desenvolvimento de um sistema capaz de controlar parâmetros musicais que, intermediado por aparatos tecnológicos, faça sentido a um atleta de fisiculturismo, usando como linguagem mimética padrões de poses previamente conhecidas e trabalhadas pelo atleta durante competições oficiais.

Na primeira etapa foi necessário estabelecer quais poses seriam usadas, qual seria a função de cada uma delas perante o sistema e qual seria a relação entre o movimento e som gerado.

Para o nível 1 de controle foram escolhidas as oito poses compulsórias¹⁰ da categoria *bodybuilding sênior* masculino determinadas pela federação IFBB Pro League Brasil, são elas: duplo bíceps; expansão de dorsais; peitoral e bíceps (peitoral melhor lado); duplo bíceps de costas; expansão de dorsais de costas; tríceps melhor lado, abdominais e coxas; mais musculoso.

Determinado os gestos de controle, iniciou-se a escolha dos elementos a serem controlados por estes comandos.

A composição foi escrita a partir dos conceitos de música interativa, exemplificado por Sweet (2015, pg. 79) como construção improvisacional de variação e forma. Estratégia composicional utilizada na criação de música para vídeo games. Sweet destaca que neste tipo de arranjo, apesar do compositor deixar à disposição várias linhas de instrumentos e sessões prontas, a decisão da estrutura musical fica por conta do interprete mediante suas ações de controle.

Baseada na estratégia de mapeamento *one-to-one mapping*¹¹ (ROVAN et al, 1997), definiu-se que a realização de cada uma das oito poses teria a capacidade de ligar ou desligar uma camada instrumental da orquestração da peça musical. Possibilitando o atleta, a partir da sequência de poses realizadas durante a performance, arranjar de diferentes modos a peça musical.

Para melhor compreensão do mapeamento gestual implementado no sistema podemos observar a tabela 1, nela está discriminado as poses de comando; sua forma de identificação gestual pelo sistema, o conteúdo sonoro mapeado para cada pose e o gesto de controle.

¹⁰ As poses compulsórias são realizadas ao mesmo tempo por grupos de atletas enfileirados em cima do palco, a fim de possibilitar uma visualização melhor pelos juízes, facilitando a comparação dos físicos.

¹¹ Na estratégia de mapeamento *one-to-one* cada gesto independente tem controle sobre um parâmetro musical. (WANDERLEY,1997)

O nível 2 foi reservado para o controle de efeitos. Após as trilhas musicais serem ativadas ou desativadas através dos comandos do primeiro nível, o atleta pode acionar o nível 2 através de um gesto de acesso, que nesse caso é a mão direita aberta (tabela 2). Neste nível há possibilidade de inserção de um filtro e dos seus controles através do movimento das mãos espelhadas no eixo horizontal.

O nível 3 possibilita o controle do volume geral da saída de áudio. Com atuação semelhante ao controle do nível 2, o terceiro nível também pode ser acessado pelo atleta através de um gesto de controle, neste caso a mão esquerda aberta (tabela 3). Após ativado, pode-se controlar o nível de sinal de volume através do movimento das mãos paralelas no eixo vertical.

Tabela 2: Mapeamento de poses do nível 1 - Controle Musical

Nome	Imagem	Mapeamento do Esqueleto				
Duplo Biceps de Frente			DBF	Discreto	Gravação de linhas de guitarras distorcidas	Tocar/Parar
Expansão de Dorsal de Frente			EDF	Discreto	Gravação de linha percussiva utilizando <i>sampler</i> de bateria acústica	Tocar/Parar
Peitoral Melhor Lado			PML	Discreto	Gravação de linha percussiva utilizando <i>sampler</i> sintetizados	Tocar/Parar
Triceps Melhor Lado			TML	Discreto	Gravação de linha melódica utilizando <i>sampler</i> de baixo sintetizado	Tocar/Parar
Abdominal e Coxa			AEC	Discreto	Gravação de linha melódica sintetizada a partir de onda quadrada	Tocar/Parar
Expansão de Dorsal de Costas			EDC	Discreto	Gravação de linhas harmônicas e melódicas utilizando <i>sampler</i> de quarteto de corda	Tocar/Parar
Duplo Biceps de Costas			DBC	Discreto	Gravação de linhas harmônicas e melódicas utilizando <i>sampler</i> de quarteto de madeiras	Tocar/Parar
Mais Musculosa			MM	Discreto	Gravação de áudio de Gongu criado a partir de <i>sampler</i> . Simultaneamente desliga todos os áudios com <i>fade out</i> de 1000 milissegundos	Desligamento geral

Tabela 3: Mapeamento de poses do Nível 2 - Controle de Efeito

Nome	Imagem	Mapeamento do Esqueleto				
Mão Direita Aberta			MDA	Discreto	Após manter durante 3 segundos esta pose é possível iniciar o controle do Nível 2	Gesto de acionamento do filtro semiparamétrico
Mão Horizontal Aberta			MHA	Contínuo	Os braços, em movimento de abdução, controlam a fc do filtro até o máximo de 8k Hz	Aumenta a fc de um filtro semiparamétrico atuante em todas as faixas de áudio ativas
Mão Horizontal Fechada			MHF	Contínuo	Os braços, em movimento de adução, controlam a fc do filtro até o mínimo de 150 Hz	Diminui a fc de um filtro semiparamétrico atuante em todas as faixas de áudio ativas

Tabela 4: Mapeamento de poses Nível 3 - Controle de Volume

Poses			Sigla	Movimento	Descrição	Gesto de Controle
Nome	Imagem	Mapeamento do Esqueleto				
Mão Esquerda Aberta			MEA	Discreto	Após manter durante 3 segundos esta pose é possível iniciar o controle do Nível 3	Gesto de acionamento do controle de dinâmica
Mão Vertical Alta			MVA	Contínuo	Os braços, em movimento de elevação, controlam em escala logarithmica o aumento do volume até 0 dB SPL	Aumenta o volume geral do sistema
Mão Vertical Baixa			MVB	Contínuo	Os braços, em movimento de descensão, controlam em escala logarithmica a diminuição do volume até -30 dB SPL	Diminui o volume geral do sistema

No nível 2 o gesto de ambos braços posicionados no centro do corpo marca a frequência mais grave de atuação do filtro. Os braços abertos em sua extensão máxima delimitam a frequência mais aguda de atuação do filtro (tabela 2). Qualquer movimento contínuo entre estes dois pontos altera o valor da frequência central (f_c) em uma escala desenvolvida sobre uma curva exponencial para controle do efeito.

Os braços para manipulação do nível 3 movimentam-se no eixo vertical. Os dois braços em paralelo levantados acima da cabeça marcam o nível máximo de volume que pode ser configurado, sendo 0 dB no sistema. Ao deslocar os braços para uma região abaixo da linha de cintura configuramos o menor valor dinâmico possível no sistema, neste caso ajustado para chegar a -30dB.

Após definido a área de intersecção em que iria se desenvolver o sistema, os parâmetros que iriam ser controlados e a estratégia de mapeamento entre gesto físico e gesto musical, deu-se início a etapa de integração de *hardwares* e *softwares*.

3.1.1 Requerimentos do Sistema

A escolha do Kinect como o sensor do sistema acarretou uma série de pré-requisitos para sua montagem. De acordo com a Microsoft¹²(2014), para que seja possível utilizar o sensor Kinect conectado a um computador, faz-se necessário que a máquina corresponda as seguintes especificações: Processador de 64 bits; Ao menos 4 GB de memória RAM; Processador com 2 núcleos reais como os modelos *dual core* de 3.1 Ghz ou superior; Portas de comunicação USB 3.0 com *chipset* da marca *Intel* ou *Renesas*; Placa de vídeo compatível com a biblioteca de *APIs* encontrada no *DirectX 11*.

Após testes de *softwares* e *hardwares*, foram escolhidos, dentre as opções que estavam ao alcance do pesquisador, as peças de menor custo e maior facilidade de reprodutibilidade. Sendo assim, foram escolhidos os seguintes aparatos:

a) Hardware

¹² Informações disponíveis no site da Microsoft, em: [https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn782036\(v=ieeb.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn782036(v=ieeb.10)). Acessada em: 12/03/2018. Uma cópia do conteúdo foi inserida na seção de apêndices para registro histórico da página web.

- computador Macbook pro com *dual boot*, para rodar o sistema operacional Windows 10; processador I7 de 2.3 Ghz Intel, 8Gb de memória RAM ddr3 com entrada USB 3.0;
- adaptador Kinect X Box One for Pc;
- Kinect Sensor modelo V2.

Figura 9: Diagrama de conexão dos hardwares constituintes do sistema BBM

b) Software

- Windows 10; (sistema operacional)
- Microsoft SDK 2.0; (Kinect *software development kit*)
- Kinect Studio; (ferramenta de gravação de vídeo)
- Visual Gesture Builder; (construtor de gestos de controle a partir de movimento)
- Visual Studio 2017; (ambiente de desenvolvimento integrado da Microsoft)
- Pure Data Vanilla 0.48.1; (ambiente de programação visual orientado a *patches*)
- Mac OSX High Sierra; (sistema operacional)
- Logic pro X 10.2. (DAW)

Figura 10: Fluxograma do sistema BBM.

3.1.2. Kinect

Dentre tantos modos de comunicação entre homem-máquina o esforço para que cada vez mais a curva de aprendizado de domínio sobre o sistema seja menor mostra-se como uma das principais preocupações dos desenvolvedores. Cardoso (2013) ressalta que após algum tempo de desenvolvimento, passando pelas interfaces CLI (*Command Line Interface*) e GUI (*Graphical User Interface*), chegamos ao conceito NUI (*Natural User Interface*), sendo que esta última nasceu para atender uma alta demanda de usuários com pouco conhecimento sobre programação, mas que desejam trabalhar com novas tecnologias de controle. Essa nova forma de interação agregou um público não especializado na área computacional, mas que por outro lado disseminou o uso dessas tecnologias por várias áreas como artes visuais, música, performance, medicina, esportes entre outros.

Juntamente com a empresa israelita PrimeSense, a Microsoft desenvolveu um hardware que combina sensor infravermelho, câmera RGB, vetor de microfone e acelerômetro (GNECCO et al, 2012.)

O projeto Natal, como foi primeiramente conhecido por causa da cidade natal de um de seus idealizadores, Alex Kipman, teve seu lançamento em novembro de 2010 (PAULA et al, 2014). Desde seu lançamento até o momento o Kinect passou por três versões, Kinect 360, Kinect for Windows e Kinect Sensor V2. A princípio o dispositivo foi criado para usuários dos consoles de videogames da Microsoft, logo ao perceber o crescente interesse de artistas e programadores a empresa liberou um pacote de ferramentas para implementação do Kinect em *softwares* de terceiros, o SDK (*System Development Kit*).

Bargsten (2014) relata que desde seu lançamento o Kinect ajudou a definir a arena do controle corporal nas artes digitais e performances, no domínio de áudio e imagem por ser um sensor de baixo custo e boa eficiência. Berger et al (2013) relata que de 2010 a 2013 mais de 3000 artigos relacionados ao Microsoft Kinect foram publicados em bibliotecas digitais: IEEE, EG, ACM, Elsevier, Springer e BMVA. Em nossa pesquisa computamos a escrita de trabalhos que utilizaram o sensor Kinect entre 2014 e 2018 nas mesmas livrarias, e encontramos 6.941 referências a utilização do sensor nos mais diversos campos de pesquisa.

O sensor Kinect 1.0 (Kinect 360) pode ser considerado uma câmera Z, por causa da sua capacidade de mapear o ambiente em três dimensões, no eixo cartesianos X, Y e Z, o que permite mensurar informação da profundidade do ambiente em tempo real interpolando imagens RGB e infravermelho e um frame rate de 30 fps. O sensor é composto de uma câmera RGB, câmera IR (*infrared*), projetor IR, um microfone *array*, motor de angulação e um acelerômetro de três eixos.

	Kinect 1.0	Kinect 2.0
RGB camera (pixel)	1280 × 1024 or 640 × 480	1920 × 1080
Depth camera (pixel)	640 × 480	512 × 424
Max depth distance (m)	4.0	4.5
Min depth distance (m)	0.8	0.5
Horizontal FOV (degrees)	57	70
Vertical FOV (degrees)	43	60
Tilt motor	Yes	No
Skeleton joint define	20	26
Full skeleton tracking	2	6
USB	2.0	3.0
Price (€)	80	199

Figura 11: Comparação dos modelos Kinect 1 e 2. (PAGLIARI et al, 2015)

Pagliari (2015) explica que o Kinect 1.0 mede a distância do ambiente através de técnica de codificação de luz. O projetor IR emite padrões de “manchas” no ambiente, a câmera IR captura a reflexão desses padrões e computa a profundidade correspondente para pixel. A medida de profundidade é feita através de processos de triangulação.

As principais características negativas do Kinect 1.0 são a baixa qualidade de informações geométrica e ruído. A câmera RGB tem pouca qualidade se comparada a webcams comuns. A imagem de profundidade também tem pouca qualidade devido ao fato do projetor IR não ter constância nos disparos dos feixes e assim muitas vezes não ser capaz de fazer a leitura do ambiente por completo.

Outra diferença entre as versões é a característica das câmeras em cada projeto. O modelo atual tem uma maior capacidade de visão horizontal de sua câmera RGB e também uma maior capacidade vertical de sua câmera IR tendo assim um alcance maior de mapeamento do ambiente.

Com o objetivo de ter melhor resolução de imagem e profundidade, uma segunda versão do sensor foi lançada com uma capacidade melhor de rastreamento do esqueleto, o Kinect 2.0.

O Kinect 2.0 tem os mesmos sensores que o Kinect 1.0, no entanto, além da otimização de seus componentes, a fórmula de medição de distância foi alterada (LACHAT et al, 2015).

Lachat et al (IDEM) relata que a medição da distância pelo sensor Kinect V2 é baseada no princípio time-of-flight¹³ (ToF), o que trouxe maior precisão para o sistema.

Ao ativar o Kinect V2, três tipos de informações são analisados. Dados infravermelho e *depthmaps* são providos de uma mesma lente e contém a mesma resolução. O mapa de profundidade (*depthmap*) é a da distância de cada pixel captado pela câmera e armazenado em 2D com 16 bits de resolução. As imagens coloridas de alta resolução são adquiridas através da câmera RGB.

¹³ Segundo Lachat et al (2015), o princípio da técnica ToF se baseia que conhecendo a velocidade da luz, a distância a ser medida é proporcional ao tempo necessário para que o raio de luz emitido pelo sensor infravermelho corra até o objeto alvo e volte ao emissor. Assim uma câmera ToF é capaz de mensurar a distância do objeto através da informação de cada pixel captado.

Os dados armazenados geram a nuvem de pontos¹⁴ graças ao mapeamento de profundidade. Através dos valores de profundidade armazenados na matrix de pixels, há duas maneiras de retirar coordenadas 3D de dados 2D. A primeira é através da função de mapeamento entre profundidade e o espaço da câmera, já implementado no SDK da companhia. Também pode-se implementar soluções externas através das relações de perspectiva de projeção. O resultado de ambos métodos será uma lista de coordenadas nos eixos X, Y e Z que podem ser utilizadas na nuvem de pontos.

Figura 12: Representação dos *outputs* de dados do Kinect V2 (LACHAT et al, 2015)

Se optar por gravar as informações colorimétricas, haverá uma transformação do quadro de cores por causa da maior resolução destes arquivos. Esta conversão também é realizada através de uma implementação nativa inclusa em seu SDK. Este mapeamento permite relacionar a cor correspondente a cada pixel no espaço amostrado. Assim o mapa de profundidade é combinado com valores de cor (RGB) e constituem uma nuvem de pontos colorida.

O Kinect V2 tem capacidade de captar até nove fontes de dados diferentes. Segundo Rahman (2017) são estes:

¹⁴ Nuvem de pontos é um conjunto de pontos expresso em um mesmo sistema de coordenadas para representar uma figura de três dimensões.

- *AudioSource*: O áudio é capturado de todas as direções, mas é distinguido em feixes que enfatize sons em uma direção específica. Os sons são armazenados em *frames*, sincronizados com o vídeo.
- *BodyFrameSource*: Armazena toda informação capturada através do mapeamento do esqueleto. Fornece coordenadas e orientações de movimento de até 6 pessoas.
- *BodyIndexFrameSource*: Fornece informação pixel a pixel de onde estão localizados o corpo e o ambiente na imagem. Útil para destacar a pessoa em frente a câmera do resto do ambiente de fundo.
- *ColorFrameSource*: Fornece dados de imagem da câmera grande angular 1080 HD do Kinect. Pode ser acessado em vários formatos de cor, como RGB e YUV. Útil para filmagem padrão e fotos.
- *DepthFrameSource*: Fornece dados de profundidade de um ambiente. A distância é dada em milímetros, medida a partir do plano da câmera para o mais próximo objeto em uma determinada coordenada.
- *InfraredFrameSource*: Através de infravermelho capta imagens de 512x424 pixels pela câmera ToF.
- *LongExposureFrameSource*: Possibilita fotografias através do sensor de infravermelho usando o mesmo ToF câmera. Possibilita gerar imagens nítidas de um primeiro plano ao desfocar o segundo plano da imagem.
- *FaceFrameSource*: Fornece o reconhecimento de 5 pontos na face em duas dimensões (coordenadas x e y). Capaz de captar alterações faciais como uma cara feliz ou triste. Útil para rastreamento facial, reconhecimento facial e adição de máscara em tempo real.
- *HighDefinitionFaceFrameSource*: Dispõe 36 pontos padrões na face e mais 600 pontos não padronizados captados em três dimensões. Utilizado para captação de micro expressões faciais e movimento do rosto para uso em animações.

O Kinect desde seu lançamento se mostra com um grande potencial para pesquisas e projetos no âmbito audiovisual. Se como controlador para jogos de videogame seu uso não desperta mais tanta comoção devido às novas tecnologias que chegam ao mercado, seu lugar em meio a diversas áreas como instrumento de mapeamento está garantido por

pelo menos mais alguns anos, principalmente por ser um eficaz complexo de sensores aliado a um baixo custo comercial.

3.2. Construindo o banco de poses

Após determinada as ferramentas que iríamos utilizar, deu-se início ao processo de estruturação do banco de poses.

Para que seja possível que o programa reconheça determinado comando, através de uma pose ou movimento, é necessário que ele tenha em seu sistema um banco de dados prévio para comparação.

Esta etapa foi realizada com dois programas inclusos no SDK 2.0 da Microsoft, *Kinect Studio* e *Visual Gesture Builder*

Kinect Studio é uma ferramenta de gravação de vídeo desenvolvida para o sensor Kinect. O software é capaz de captar os nove fluxos de dados, discriminados no tópico anterior, individualmente, e de armazenar metadados junto com o arquivo de vídeo.

O programa *Visual Gesture Builder* é capaz de processar o formato de arquivo criado pelo *Kinect Studio*, lendo todo os metadados armazenados. É neste programa que ocorre a delimitação de cada movimento selecionado para ser gesto de controle, e através do processo de *machine learning*¹⁵ cria-se o pacote de dados contendo as coordenadas de cada movimento que serão utilizados como biblioteca para comparação durante a utilização do sistema.

3.2.1 Captação do banco de poses

A fase de captação se deu pelo programa Kinect Studio. Segundo Rahman (2017, p 33):

¹⁵ “*Machine Learning* é um método de análise de dados que automatiza a construção de modelos analíticos. É um ramo da inteligência artificial baseado na ideia de que sistemas podem aprender com dados, identificar padrões e tomar decisões com o mínimo de intervenção humana.” (SAS INSTITUTE, 2018)

Kinect Studio é um aplicativo fornecido junto ao SDK 2.0 que permite capturar e reproduzir dados coletados pelo Kinect. Você pode muito bem filmar uma cena com o Kinect, incluindo as várias fontes de dados e assistir em algo como um player de vídeo. Estes dados podem ser enviados para as suas aplicações, em vez de ter que gerar filmagens para cada aplicação. Isso não apenas reduz gestos repetitivos e cansativos, mas também permite configurar controles para seus experimentos algorítmicos para que assim você possa determinar as diferenças exatas em seu desempenho¹⁶. (tradução nossa)

Ferramenta desenvolvida e disponibilizada desde o primeiro SDK pela Microsoft, o Kinect Studio tem função de facilitar a implementação de movimentos de controle através da captura da imagem em conjunto com os metadados de mapeamento.

Figura 13: Aba de leitura de vídeo do Kinect Studio.

Na figura 13 pode-se ver as informações armazenadas em um arquivo de vídeo capturado através do sensor Kinect. A figura a direita é a captura da imagem em *colored point cloud*, junto com rastreamento do esqueleto e consecutivamente das juntas do corpo. Abaixo há discriminado as fontes de dados que selecionamos para serem gravadas.

¹⁶ “Kinect Studio is an application bundled with the SDK that allows you to capture and replay data collected by the Kinect. You can pretty much film a scene with the Kinect, including all the various Data Source, and watch it in something sort of like a video player. This data can be fed to your Kinect applications instead of your having to generate footage each time around. Not only does this reduce repetitive, tiring gestures, but it also allows you set up controls for your algorithmic experiments so that you can determine each difference in their performance.”

Figura 14: Imagem captada em *colored cloud point* pelo sensor Kinect

Para criação de nosso banco de poses selecionamos 8 das 11 *streams*¹⁷ de dados disponíveis para serem capturados pelo sensor.

Figura 15: Opções de *streams* selecionadas na captura de vídeo das poses do sistema BBM.

¹⁷ Forma como é referenciado pelo programa Kinect Studio as diferentes fontes de dados possíveis de serem captados pelas câmeras e microfone do sensor Kinect

As *streams* que não foram selecionadas são: *Nui Title Audio*, *Nui Uncompressed Color* e *Nui Color Camera Settings*. O motivo para escolha de não captar estes parâmetros foram: primeiro a não utilização de dados provenientes do áudio e da imagem RGB para controle do sistema interativo, segundo a diminuição do tamanho dos arquivos, que chegam a ter 1 Gigabyte de informação para 30 segundos de vídeo.

Nome	Data de modificação	Tipo	Tamanho
Joao	05/07/2018 11:08	Pasta de arquivos	
Vinicius	05/07/2018 11:14	Pasta de arquivos	
DBF 5.vgbcclip	13/07/2018 16:04	Arquivo VGBCLIP	42 KB
DBF 5.xef	10/07/2018 16:14	Kinect Studio	640.091 KB
DBF 6.vgbcclip	06/08/2018 21:30	Arquivo VGBCLIP	20 KB
DBF 6.xef	10/07/2018 16:15	Kinect Studio	608.781 KB
DBF 7.vgbcclip	13/07/2018 16:05	Arquivo VGBCLIP	38 KB
DBF 7.xef	10/07/2018 16:25	Kinect Studio	798.129 KB
DBF 8.vgbcclip	06/08/2018 21:30	Arquivo VGBCLIP	22 KB
DBF 8.xef	10/07/2018 16:26	Kinect Studio	578.962 KB
PML 5.vgbcclip	12/07/2018 22:44	Arquivo VGBCLIP	25 KB
PML 5.xef	10/07/2018 16:16	Kinect Studio	912.932 KB
PML 6.vgbcclip	10/07/2018 16:47	Arquivo VGBCLIP	34 KB
PML 6.xef	10/07/2018 16:17	Kinect Studio	932.313 KB
PML 7.vgbcclip	30/08/2018 11:55	Arquivo VGBCLIP	25 KB
PML 7.xef	10/07/2018 16:24	Kinect Studio	842.858 KB
PML 8.vgbcclip	10/07/2018 16:48	Arquivo VGBCLIP	30 KB
PML 8.xef	10/07/2018 16:25	Kinect Studio	821.984 KB
TML 5.vgbcclip	30/08/2018 11:58	Arquivo VGBCLIP	38 KB
TML 5.xef	10/07/2018 16:17	Kinect Studio	1.264.791 KB
TML 6.vgbcclip	30/08/2018 11:57	Arquivo VGBCLIP	38 KB
TML 6.xef	10/07/2018 16:19	Kinect Studio	1.109.735 KB
TML 7.vgbcclip	10/07/2018 16:54	Arquivo VGBCLIP	37 KB
TML 7.xef	10/07/2018 16:23	Kinect Studio	1.117.189 KB

Figura 16: Uma das quatro pastas de arquivos com vídeos gravados pelo Kinect Studio para exemplificar o tamanho dos arquivos de imagem.

Para gravar as poses utilizou-se dois atletas¹⁸ adeptos do esporte e participantes de campeonatos de fisiculturismo. Foi feita a escolha por atletas do ramo como atores para que as poses fossem feitas mediante as regras estabelecidas nos campeonatos. No caso foram seguidas as diretrizes da IFBB internacional em relação as poses mandatórias da categoria *bodybuilding* sênior¹⁹, as quais são usadas como gestos de controle na implementação deste sistema.

O programa *Visual Gesture Builder* (VGB) utiliza os metadados inclusos nos vídeos captados pelo programa Kinect Studio para comparação de posições cartesianas do corpo presente no vídeo. Através de algoritmo de estatística avançada, VGB analisa

¹⁸ Vinicius Ramos: Campeão IFBB estreantes e paulista 2012; campeão paulista e 3º colocado campeonato brasileiro 2014 IFBB; 2º colocado no campeonato brasileiro e 16º colocado no campeonato mundial 2015; 10º colocado no Arnold *South America* 2016.

João Monnazzi: 5º colocado campeonato estreantes IFBB 2013; 7º colocado campeonato paulista IFBB 2014; 3º colocado campeonato paulista e 10º colocado campeonato brasileiro IFBB 2015.

¹⁹ Uma cópia da última resolução oficial das poses mandatórias está no apêndice desta dissertação.

os pontos coincidentes mapeados e cria o modelo do gesto que deve ser escaneado. Para que haja uma maior precisão no rastreamento das poses, alguns gestos necessitaram de mais cliques para análise que outros.

Ao todo foram gravados 71 vídeos de poses para criação da biblioteca de poses para o sistema. O movimento de cada pose foi repetido três vezes em cada vídeo gravado.

O nível 1 de controle contou com 47 vídeos de poses para sua estruturação. Para as poses AEC, DBC, EDC, EDF e MM foram captados 4 vídeos para cada. Para TML 7, PML 8 e DBF 10 cliques foram gravados. Ainda se fez necessário a gravação de 2 vídeos alternando entre as poses EDF/EDC e DBF/DBC. Estes últimos foram necessários para a marcação negativa dos movimentos, sendo que a mesma pose feita de frente e de costas devem ser interpretadas como poses distintas.

Para o nível 2 de controle foram gravadas 8 poses com os braços em diferentes alturas no eixo vertical para que este valor fosse desconsiderado, apenas sendo considerado para o mapeamento, o movimento dos braços no eixo horizontal.

O número de cliques gravados para o nível 3 de controle foi o mesmo que o nível anterior, usando a mesma metodologia, para que no mapeamento do movimento horizontal fosse desconsiderado, gravou-se gestos com os braços em diferentes aberturas horizontal, porém marcando apenas os valores de movimento no eixo vertical.

Houve a captação de posições de mãos que implementariam o nível 0, nível que guardaria as poses de acionamento e desligamento dos níveis 2 e 3. Porém a biblioteca nativa, que é distribuída junto ao SDK, já contém os estados de mão mapeados por padrão, não havendo necessidade de mapear estes gestos para desenvolver uma biblioteca.

3.2.2 Construção de uma biblioteca de poses a partir de gestos captados

Após a etapa de captação dos vídeos com as poses vem o processo de marcação dos movimentos que devem ou não ser considerados válidos pelo sistema. Esta etapa foi realizada no programa *Visual Gesture Builder*.

Visual Gesture Builder (VGB) é uma ferramenta para implementações do Kinect em sistemas operacionais Windows, que usa algoritmos de *machine-learning* para criar banco de dados gestuais. Estes bancos de dados são usados para detecção em tempo real

através da API²⁰ do VGB. Após a captura dos vídeos pelo Kinect Studio, VGB usa o algoritmo *AdaBoost*²¹ para determinar quando o usuário performa um certo gesto e o algoritmo *RFRProgress*²² para determinar a progressão de um gesto (RAHMAN, 2017).

Nosso sistema conta com um banco de dados tanto de gestos discretos, quanto de gestos contínuos.

Para controle do nível 1 é utilizado detecção de poses discretas. Foram mapeados para este nível as 8 poses compulsórias da categoria fisiculturismo masculino sênior. Neste nível para que o atleta possa controlar o acionamento ou desligamento de determinado objeto sonoro é necessário a realização de uma das 8 poses em frente o Kinect.

O mapeamento deste nível se deu através da marcação pose por pose no VGB. O processo se resume a lançar os cliques captados pelo Kinect Studio e marcar no vídeo os frames que condizem com a pose que queira que seja identificada.

Como o VGB trabalha com dados estatísticos para avaliar o que deve ou não ser entendido como gesto, quanto mais vídeos, com pessoas e ambientes diferentes forem jogados no repositório e marcados, haverá um banco de dados maior para que a análise estatística seja feita, o que tornará a compreensão do gesto mais fácil pela implementação programada. Para este nível foram gravados e analisados 47 vídeos para criação do banco de poses.

²⁰ *Application Programming Interface*: Conjunto de rotinas e padrões de programação cedidos pelo fabricante para que a implementação de seu programa a outras aplicações seja mais fácil.

²¹ *Adaptive Boosting*. Algoritmo classificador binário usado para identificação de gestos discretos. Ou o gesto está sendo feito (1), ou não está (0). (RAHMAN, 2017, p. 228)

²² *Random Forest Regression Progress*. Algoritmo para identificação de posição durante um gesto contínuo. O valor de progressão é 0 quando está no início e 1 quando chega em seu estágio final. Qualquer posição entre esses dois pontos é determinada por um valor oscilante.

Figura 17: Marcação das poses discretas no VGB para a construção do repositório

O primeiro passo para criação do banco de dados é criar no programa Visual Gesture Builder, um repositório, que neste programa é denominado como *solutions*. Neste repositório será colocado todos os vídeos, separados por gestos, que serão usados para determinada aplicação.

Para este projeto, foi necessário a criação de três repositórios, um para cada nível, sendo que suas aplicações atuariam em locais diferentes.

Figura 18: Repositórios criados no VGB para o banco de poses dos níveis 1,2 e 3

Dentro de cada repositório há duas pastas. A primeira, denominada *training project*, é onde se armazena os vídeos e ocorre a marcação das poses. A segunda pasta, denominada *analysis project*, é usada para testar a eficácia das marcações feitas e o entendimento que está sendo obtido de cada gesto.

No VGB a marcação dos frames válidos para caracterização da pose se dá na aba *gesture tag*, localizada logo abaixo dos players de vídeo (figura 19). O que está delimitado de azul no canto superior da barra, o programa armazena como informação gestual válida. As áreas demarcadas em azul no campo inferior da barra são consideradas como informação negativa e que não fazem parte do gesto. Áreas não delimitadas também são entendidas como informações negativas que não devem ser consideradas.

A marcação de gestos contínuo requerem duas etapas. A primeira consiste na delimitação do movimento inicial e final através da marcação de dois gestos discretos.

Figura 19: Marcação do gesto discreto inicial do movimento

Figura 20: Marcação do gesto discreto final do movimento

A segunda etapa consiste na marcação do gesto contínuo marcando onde o gesto terá o valor mínimo (igual a 0), onde terá valor máximo (igual a 1) e como será representada a escala de progressão entre os dois pontos (figura 21)

As marcações de gestos contínuo foi necessário para o controle do nível 2 (controle de efeito) e nível 3 (controle de dinâmica).

Figura 21: Marcação do gesto contínuo entre o ponto de início e de término.

O maior desafio nesta etapa foi a delimitação das poses do nível 1. As semelhanças entre algumas poses como as poses DBF e DBC, EDF e EDC, sendo que são praticamente a mesma pose, apenas uma feita de frente e a outra de costas, exigiram um número de vídeos maior para marcações, inclusive fazendo uso da marcação negativa em algumas poses. Ou seja, a colocação de determinado vídeo marcado totalmente como negativo dentro de uma solução para diminuir as proporções percentuais de gestos que não devam ser captados.

Após marcados todos os clipes dentro de cada repositório os dados foram compilados, gerando um arquivo de banco de dados (extensão **.gbd**), contendo coordenadas espaciais de cada ponto do *bodyframe* que deve ser analisado e comparado pela aplicação para que haja reconhecimento dos gestos marcados.

3.3. Programação do *Middleware*: transcritor coreográfico-musical e montador de mensagens OSC

Para que fosse possível a utilização do Kinect em tempo real no sistema, foi necessário desenvolver um programa que armazenasse o banco de dados criados, processasse as informações recebidas através do sensor, comparasse as informações com o banco de dados previamente criado e enviasse mensagens de comandos através de um protocolo de comunicação que outro programa de manipulação musical fosse capaz de decodificar.

Como todos os programas da Microsoft que foram usados são programados na linguagem C#, optou-se por montar o *middleware* na mesma linguagem, afim de evitar conversões em outros estados do processo.

O executável é desenvolvido em três módulos, que desempenham 5 funções. A primeira é armazenar nosso banco de dados gestual desenvolvido previamente. A segunda é receber os dados do sensor Kinect em tempo real. Decodificar os códigos e mapear os dados recebidos para as funções programadas é a terceira e quarta funções programadas e, por último, a quinta função é enviar os dados obtidos para o gerador musical via protocolo de comunicação OSC.

Esta parte do projeto foi desenvolvida no programa Microsoft Visual Studio e contou com o auxílio do programador Carlos Gabriel Luz Monnazzi. O aplicativo foi denominado de KG_OSC²³ (*Kinect gesture to OSC message*) e, como sugere o nome, perfaz o mapeamento dos gestos captados pelo Kinect para mensagens OSC de controle para o gerador musical.

Figura 22: Programação da ferramenta *Kinect_Gesture to OSC* no Microsoft Visual Studio

Usamos ao total cinco bibliotecas na implementação deste programa:

1. *Microsoft.Kinect.dll*, disponível no SDK 2.0 que traz dados gerais para implementação de aplicativos que irão usar o Kinect.
2. *Microsoft.Kinect.VisualGestureBuilder.dll*, também distribuída através do SDK da empresa, tem a função de traduzir os dados que foram compilados

²³ O código programado para esta função pode ser acessado e baixado livremente por desenvolvedores através do repositório: https://github.com/dreamblader/KG_OSC.

pelo VGB para que os arquivos **.gbd** possam ser acessados como repositório de gestos.

3. Uma biblioteca desenvolvida por terceiros, denominada VentuzOSC²⁴, já com a programação para endereçar dados produzidos pelo Kinect para uma saída de rede (OSC), possibilitando o uso dos dados para controlar outra aplicação em tempo real.

E por fim as bibliotecas *AdaBoostTech.dll* (4) e *RFRProgressTech.dll* (5), para identificação e tradução em tempo real dos dados adquiridos através do rastreamento de esqueleto, possibilitado pela análise das informações captadas de *body frame rate* e *joint detection* pelo Kinect.

O código completo do aplicativo *Kinect_Gesture to OSC* está mostrado na subseção a seguir.

3.3.1 – Classe de iniciação do middleware

A classe *Main_App*, mostrada à figura 23, serve para iniciar o programa *middleware*.

²⁴ VentuzOSC é distribuída de forma gratuita para desenvolvedores de aplicações que necessitem usar o protocolo de comunicação OSC. A biblioteca pode ser acessada em: http://www.ventuz.com/download/public/OSC/V1_4_1/Ventuz.OSC.chm

```

{
    class Main_App
    {
        static void Main()
        {
            //main software runner
            bool running = true;

            Translator main_translate = new Translator();

            while(running)
            {
                if (Console.KeyAvailable) //get the key from a
stream stack, dont block the app waiting for a key press
                {
                    ConsoleKeyInfo exit_command =
Console.ReadKey(true); // key command listener

                    if (exit_command.Key == ConsoleKey.Enter)
                    {
                        running = false;
                        //Console.WriteLine("KILLER"); << DEBUG
PRINT
                    }
                }

                // run listener (transformer function)
            }
        }
    }
}

```

Figura 23: Programa de iniciação do middleware do sistema BBM.

3.3.2 *Gesture Detector*

Esta classe é responsável por analisar os gestos e criar mensagens de controle através destes. Foi nesta etapa do processo que agregamos o repositório de poses criado previamente como uma biblioteca, aqui denominada *Gesture_Library*. Definimos o comando gerado através do reconhecimento do gesto e criamos os pacotes de mensagens para serem enviados via OSC.

O cabeçalho do código carrega as bibliotecas de gestos dos 3 níveis, deixando prontas para serem usadas para comparação enquanto o Kinect estiver ligado. Logo após o carregamento dos bancos de poses, é configurado um *gate* para que a pose seja validada apenas se tiver um nível de semelhança (no código é denominado como *confidence*) na comparação entre gesto e pose de mais de 85% e que este nível seja mantido por 10 frames consecutivos, o que dá em torno de 1/3 de segundo.

A segunda etapa deste processo habilita as bibliotecas inclusas no SDK 2.0 que permitem a análise de poses através das informações captadas.

O último passo do código configura as condições que devem ser alcançadas para que o reconhecimento de uma pose vire uma informação de comando e qual informação será enviada por OSC mediante ao reconhecimento de cada pose e mapeia estas informações para valores numéricos.

Para que uma pose seja ativada, como já foi explicado previamente, é necessário que o gesto realizado em tempo real e as informações contidas no repositório de pose tenham um grau de confiança maior que 85% por mais de 10 *frames* consecutivos. O grau de confiança deve cair a menos de 65% pelo os mesmos 10 *frames* consecutivos para que a pose deixe de ser considerada ativa.

As poses do nível 1 de controle, enviam dois pacotes de dados sendo mapeados da seguinte forma.

1. *Id Track (identification track)*: Os níveis de controle aqui descritos foram referenciados por *type* no KG_OSC. Sendo assim o *type* 1 descritos nos códigos abaixo se refere ao nível 1. Quando acionado o nível 1 envia-se a mensagem contendo o número 1, para designar a qual dos níveis se destina a mensagem.
2. *OSC_adress*: A segunda informação enviada é o número de endereço da mensagem. No KG_OSC, cada pose recebeu um número como código de envio. Assim quando ocorre a identificação de determinada pose, o KG_OSC envia o número referente daquela pose para o gerador musical. Este por sua vez tem em seu sistema um roteador que identifica o número e o envio para o local correto.

Para o nível 2 de controle, foram reservados três pacotes de mensagens de controle.

1. *Id Track (identification track)*: A primeira mensagem, como no nível 1, é reservado para designar qual nível será controlado. No caso do nível 2, envia-se o número 2 como mensagem.
2. *Id_fx*: O segundo pacote foi criado pensando em aplicações futuras e está reservado para o número do *Id Track*. Com esta aplicação, torna-se possível a inserção e manipulação de efeitos em objetos sonoros separados. Como este recurso não está sendo usado no momento, a

mensagem está sendo usada para acionar e desligar o filtro enviando uma mensagem com o número -1.

3. *Fx_Value*: mensagem de fluxo contínuo de informação onde são enviados valores de 0 a 127 de acordo com os movimentos do braço do usuário. No nível 2, o valor 0 remete a menor frequência central em que o filtro pode atuar, aqui configurado para 180 Hz. O valor 127 marca a maior frequência de atuação possível, aqui delimitado em 12 KHZ. Qualquer valor entre esses dois números pode ser enviado.

O nível 3 de controle tem basicamente a mesma programação que o nível 2 sendo no código chamado por *type 3*.

1. *Id Track*: A primeira mensagem configura o nível que será controlado. Neste caso é enviado a mensagem com o número 3.
2. *Id_fx*: O segundo pacote está reservado para aplicações futuras, podendo ser designado o controle de volume de objeto sonoros individualmente.
4. *Velocity*: mensagem de fluxo contínuo de informação onde são enviados valores de 0 a 127 de acordo com os movimentos do braço do usuário. No nível 3, o valor 0 remete ao menor volume que o processador pode configurar, aqui -30 dB. O valor 127 marca o maior volume possível, aqui delimitado em 0 dB. Qualquer valor entre esses dois números pode ser enviado.


```

namespace Kinect_Gesture_to_OSC
{
    class GestureDetector
    {
        // Class for loading gestures and creating objects from them

        //private static List<Gestures> Gesture_Library = new List<Gestures>(); //gesture library (list of all
created gestures)
        // VGB Databases
        private readonly string type1_database = @"Gestures\Type 1.gbd";
        private readonly string type2_database = @"Gestures\Type 2.gbd";
        private readonly string type3_database = @"Gestures\Type 3.gbd";

        private Gesture gesture_history = null;
        private int result_score = 0; //score to get enough time to trigger osc message (10 frames = 1 third of
second)
        private bool result_cooldown = true; //cooldown for result score timer

        private int type2_init; //position of start of all type 2 gestures inside VgbFrameSource
        private int type3_init; //position of start of all type 3 gestures inside VgbFrameSource

        private bool Type2_gate;
        private bool Type3_gate;

        /// <summary> Gesture frame source which should be tied to a body tracking ID </summary>
        private VisualGestureBuilderFrameSource vgbFrameSource = null;

        /// <summary> Gesture frame reader which will handle gesture events coming from the sensor </summary>
        private VisualGestureBuilderFrameReader vgbFrameReader = null;

        private OSC_Messages osc_message = null; // final osc message

        public GestureDetector(KinectSensor sensor) //constructor
        {
            //var gesture = Gesture_Loader(gesture_name);

            this.vgbFrameSource = new VisualGestureBuilderFrameSource(sensor, 0);
            this.vgbFrameReader = this.vgbFrameSource.OpenReader();
            if (this.vgbFrameReader != null)
            {
                IsPaused = true;
                this.vgbFrameReader.FrameArrived += this.Gesture_Reader;
            }

            // VGB Gestures Loader
            using (VisualGestureBuilderDatabase database = new VisualGestureBuilderDatabase(this.type1_database))
            {
                // we could load all available gestures in the database with a call to
vgbFrameSource.AddGestures(database.AvailableGestures),
                this.vgbFrameSource.AddGestures(database.AvailableGestures);
            }

            type2_init = vgbFrameSource.Gestures.Count; // marks the beggining of the type 2

            //TYPE 2 DB
            using (VisualGestureBuilderDatabase database = new VisualGestureBuilderDatabase(this.type2_database))
            {
                // we could load all available gestures in the database with a call to
vgbFrameSource.AddGestures(database.AvailableGestures),
                this.vgbFrameSource.AddGestures(database.AvailableGestures);
            }

            type3_init = vgbFrameSource.Gestures.Count; // marks the beggining of the type 3

            //TYPE 3 DB
            using (VisualGestureBuilderDatabase database = new VisualGestureBuilderDatabase(this.type3_database))
            {
                // we could load all available gestures in the database with a call to
vgbFrameSource.AddGestures(database.AvailableGestures),
                this.vgbFrameSource.AddGestures(database.AvailableGestures);
            }
            //-----
        }
    }
}

```

[illegible]

```

discreteResults.TryGetValue(gesture, out result);

high confidence
if (result.Detected && result.Confidence > 0.85) //got a discret result with a
{
    //Console.WriteLine(result_score); //DEBUG TEST SCORE

//Create OSC Message HERE (?)
    if(gesture_history != gesture)
    {
        gesture_history = gesture;
        result_score = 1;
        result_cooldown = true;
    }
    else if(gesture_history == gesture)
    {
        result_score++;
        if (result_score >= 15) //soft cap the score maximum to 15
        {
            result_score = 15;
        }
    }

    if(result_score > 10 && result_cooldown)
    {
        Console.WriteLine("A pose: " + gesture.Name + " foi efetuada com
sucesso.");
        Gesture_List_to_OSC(gesture);
        result_cooldown = false;
    }
}

else if (result.Confidence < 0.6 && gesture == gesture_history) //if a gesture
saved in the history is not anymore being tracked >>> lose score. If score reach 0 the position is considered
not being tracked anymore
{
    result_score--;

    if(result_score<=0)
    {
        gesture_history = null;
        result_cooldown = true;
        result_score = 0; //redundancy to be extra sure that the score is
setted to 0
    }
}

i++;
}

//-----
//-----

if (continuousResults != null)
{
    i = 0; // reset i counter

    foreach (Gesture gesture in this.vgbFrameSource.Gestures)
    {
        bool type_positive = false; // see if the type gate and requirements are met

        if (gesture.GestureType == GestureType.Continuous)
        {
            if(Type2_gate && (i >= type2_init && i < type3_init)) //if the gate is open
and the gesture intervals meet the type 2 inside vgbFrameSource list
            {
                type_positive = true;
            }
            if(Type3_gate && (i >= type3_init)) //if the gate is open and the gesture
intervals meet the type 3 inside vgbFrameSource list
            {
                type_positive = true;
            }
        }

        if(type_positive)
        {
            ContinuousGestureResult result = null;
            continuousResults.TryGetValue(gesture, out result);

            if (result != null)
            {

```

```

        Gesture_List_to_OSC(gesture, result.Progress);
    }
}

    }
    i++;
}
}
}
}

private void Gesture_List_to_OSC(Gesture user_gesture, float continous_progress = -1) // function to
compare which gesture got triggered and create a OSC Message based on it
{
    double converted_value_type2 = (continous_progress * 127); //Value of conversion will be 0 [min] to
127 [max]. MIDI values (for type 2 messages)
    double converted_value_type3 = (continous_progress * 116) + 11; //Value of conversion will be 11 [min]
to 127 [max]. Db values (for type 3 messages)
    converted_value_type2 = Math.Round(converted_value_type2);
    converted_value_type3 = Math.Round(converted_value_type3);

    if(user_gesture == null) // hand gesture - only gesture exception
    {
        osc_message = new OSC_Messages(2, -1, 1, 0); //liga ou desliga o filtro (id_fx = 1)
    }
    else
    {
        switch (user_gesture.Name)
        {
            case "Duplo_biceps_frente":
                osc_message = new OSC_Messages(1, 1);
                break;

            case "expansao_de_dorsal":
                osc_message = new OSC_Messages(1, 2);
                break;

            case "Peitoral_melhor_lado":
                osc_message = new OSC_Messages(1, 3);
                break;

            case "triceps_melhor_lado":
                osc_message = new OSC_Messages(1, 4);
                break;

            case "Abdominal_e_Coxa":
                osc_message = new OSC_Messages(1, 5);
                break;

            case "Expansao_de_dorsal_costas":
                osc_message = new OSC_Messages(1, 6);
                break;

            case "Duplo_biceps_costas":
                osc_message = new OSC_Messages(1, 7);
                break;

            case "Mais_musculosa":
                osc_message = new OSC_Messages(1);
                break;

            case "FiltroProgress":
                osc_message = new OSC_Messages(2, -1, -1, (float)converted_value_type2); // id_fx = -1 so
it can't effect the on/off switch of the filter
                break;

            case "volume_minProgress":
                osc_message = new OSC_Messages(3, -1, -1, -1, (float)converted_value_type3);
                break;

            default:
                break;
        }
    }

    if(osc_message != null)

```

```

    }
    if(osc_message != null)
    {
        osc_message.Send_OSC();
    }
}
}
}

```

Figura 24: Código C# da seção de reconhecimento de gestos do *Kinect_Gesture to OSC* no Microsoft Visual Studio

3.3.3 Transmissor OSC

Nesta etapa para que houvesse comunicação entre o programa de armazenamento do banco de poses, controle do Kinect e Pure Data (PD), foi escolhido o protocolo de comunicação OSC²⁵

Após termos mapeados cada pose correspondendo com um valor numérico na classe *gesture_detector* para ser enviado ao programa de geração musical, agora é a vez de criar o arquivo que será enviado.

Para se enviar as mensagens de comando foi usada a ferramenta VentuzOSC. Esta biblioteca já contém previamente estabelecidos padrões para o envio de mensagens via OSC a partir de dados adquiridos pelo Kinect *sensor v2*.

A primeira configuração que foi feita neste processo foi a definição do *ip* para 127.0.0.1, endereço padrão de comunicação para conexões entre programas diferentes, mas internos a um mesmo sistema, e das portas de saída e entrada de dados, para que ambos programas estejam conectados no mesmo canal. A porta configurada foi a 9001, porta de comunicação padrão do Pure Data Vanilla.

Por fim definiu a ordem que as mensagens devem ser mandadas a fim que o decodificados do PD receba-as de forma correta, ficando da seguinte forma.

1. *Int type*: Envia um número fixo entre 1 e 3, definindo qual nível está sendo requisitado.
2. *Int id_track*: Envia um número fixo entre 1 e 8, definindo qual pose foi acionada.
3. *Float fx_value*: Envia continuamente fluxo de dados numéricos entre 1 e 127 para controle do processador de efeito quando o nível 2 está acionado.

²⁵ *Open Sound Control* é um protocolo de rede de parâmetros flexíveis usado para transmissão de dados entre aplicações musicais.

4. *Float velocity*: Envia continuamente fluxo de dados numéricos entre 1 e 127 para controle do processador de dinâmica quando o nível 3 está acionado.

Por último são configuradas as mensagens que serão escritas no *prompt* de comando para que o usuário saiba que a conexão foi bem-sucedida. O código desta parte do programa está descrito na figura 25

```

class OSC_Messages
{
    //class for creating OSC default messages to send to other apps.
    private int Id_track;
    private int Id_fx;
    private float Fx_value;
    private float Velocity;
    private string osc_address;

    // Load an OSC DLL and set it Objects -- VentuzOSC
    private static string ip_address = "127.0.0.1";
    private static int port_num = 9001;
    private OscBundle osc_pack = new OscBundle(); // Bundle Object -- used to hold all OSC INFO
    (OscElements) to be transported using the UDPWriter.
    UdpWriter udp_obj = new UdpWriter(ip_address, port_num); // Object used to Write/Send OSC
    Packages using UDP protocol.

    // -- Default Package Info ---
    // Separated in 3 Types of address and constructors overloads << unused for now
    // INT - generate the address string for OSC type using right now: \type1, \type2 n \type3
    (based on the movement type section) // INT - for ID of Track that will be played // INT - for ID of FX
    that will be used // FLOAT - for value that will be applied in the FX // FLOAT - for value 8applied in
    the Velocity factor of the song //
    public OSC_Messages(int type, int id_track = -1, int id_fx = -1, float fx_value = -1, float
    velocity = -1) //constructor for Type 1 Command - Only Affect Tracks - << unused for now
    {
        Id_track = id_track;
        Id_fx = id_fx;
        Fx_value = fx_value;
        Velocity = velocity;
        osc_address = @" /type" + type; //OSC Address
        osc_pack.AddElement(Pack_OSC()); // Call Pack_OSC function creating the OSCElement to be
        added to the OSCBunble
    }

    // BETTER LET 1 CONSRUCTOR WITH THE ADDRESS ON IT AND DEFAULT VALUES AT -1
    /*
    public OSC_Messages(int id_fx, float fx_value) // contructor for Type 2 Command - Only Affect
    FXs and they Values -
    {
        Id_fx = id_fx;
        Fx_value = fx_value;
        osc_address = "\type2"; //Address - /type2
        Pack_OSC();
    }

    public OSC_Messages(float velocity) // contructor for Type 3 Command - Only Affect Velocity -
    {
        Velocity = velocity;
        osc_address = "\type3"; //Address - /type3
        Pack_OSC();
    }
    */

    private OscElement Pack_OSC() //used to pack OSC Elements independent of the constructor
    (multiple constructors not used, but function still implemented for future changes)
    {
        OscElement osc_obj = new OscElement(osc_address, null);

        if (osc_address == @" /type1")
        {
            osc_obj = new OscElement(osc_address, Id_track);
        }
        else
        {
            if (osc_address == @" /type2")
            {
                osc_obj = new OscElement(osc_address, Id_fx, Fx_value);
            }
            else
            {
                if (osc_address == @" /type3")
                {
                    osc_obj = new OscElement(osc_address, Id_track, Velocity);
                }
                else // no real type assigned: Return Error X and kill the application
                {
                    Console.WriteLine("ERROR X: No real type assigned (only supporting types: 1, 2 or 3)");
                    Console.WriteLine("FOR MORE INFO READ THE GUIDE.TXT IN THIS APP ROOT DIRECTORY");
                    Console.Read(); // just a "pause" for reading printed stuff
                    Environment.Exit(-1);
                }
            }
        }

        return osc_obj;
    }
}

```

```

    }

    // Default Package is -1. This value implies that this feature will be disabled at the end
    software wich receive this OSC Message

    //Prototype: (messages, UDP_port) >>>> maybe more?
    public void Send_OSC() //sending function
    {

        Console.WriteLine("Connection IP: " + ip_address + " Port:" + port_num);
        Console.WriteLine("OSC Address: " + osc_address);
        Console.WriteLine("This OBJ Variables: " + Id_track + " " + Id_fx + " " + Fx_value + " " +
Velocity);
        Console.WriteLine("Number of Elements in the Package: " + osc_pack.Elements.Count);
        Console.WriteLine("\n");

        //troubleshoot UDP SUCESS??? (is this possible?)
        udp_obj.Send(osc_pack);
    }

    ~OSC_Messages() //destructor
    {
        udp_obj.Dispose();
    }
}
}

```

Figura 25: Código C# do módulo Transmissor OSC, que perfaz a programação de comunicação para envio de mensagens entre o Kinect e o Pure Data através do protocolo OSC

3.3.4 Translator

A classe que chamamos de *translator* tem a função de configurar as análises que serão realizadas entre o banco de poses e a mensagem que o Kinect está captando em tempo real. Nesta seção delimitamos o uso do sistema por apenas um usuário por vez, sendo que se houver duas pessoas na zona de visão do Kinect o sistema entrará em *stand by*.

As variáveis de dados que foram configurados para comparação são: *CoordinateMapper*; *BodyFrameSource* e *GestureDetector* (figura 26), responsáveis por calcular as coordenadas que estão sendo captadas, o rastreamento do esqueleto através dos *frames* e comparar com o banco de poses criado.

Após a comparação entre os dados de poses, estados de mão que estão sendo recebido e os dados no repositório, define-se a ação que deve ser tomada a partir do reconhecimento de cada gesto.


```

class Translator //class for capturing gestures and translate them into OSC messages
{
    private KinectSensor sensor = null; // Sensor
    private Body[] bodies = null; // Multiple Bodies tracker
    private BodyFrameReader bodyFrameReader = null; // Reader for body frames
    private CoordinateMapper coordinatemapper = null; // Coordinate mapper to map one type of point to
another
    private int body_history = -1; // Maintain last number of bodies captured - used to not print the same
message again if number of bodies didn't change
    //private OSC_Messages osc_message = null; // final osc message -- will be created if a gesture is
triggered. ----- TRANSFERED TO GESTURE DETECTOR CLASS -----
    private bool it_can_compare = false; // see if the conditions of comparing the body have met (1 body
in screen)

    private GestureDetector gestureDetector = null; // Gesture Detector object to be used in the gesture
detection

    // HANDSTATES VARIABLES
    private HandState right_hand_consistency = new HandState(); // used to see if last right hand state
didn't change
    private HandState left_hand_consistency= new HandState(); // used to see if last left hand state
didn't change
    private int right_hand_score = 0; // this right hand "hold" time
    private int left_hand_score = 0; // this left hand "hold" time
    private bool right_cooldown = false; // cooldown so the user don't change states just by holding his
hand longer
    private bool left_cooldown = false; // /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\
/\ /\ /\ /\ /\ /\
    // -----
    -----

    bool type2_gate = false; // condition that make the type 2 list starts to being compared
    bool type3_gate = false; // condition that make the type 3 list starts to being compared

    public Translator() //constructor
    {
        //setting translator Kinect variables
        this.sensor = KinectSensor.GetDefault();
        this.coordinatemapper = this.sensor.CoordinateMapper;
        this.bodyFrameReader = this.sensor.BodyFrameSource.OpenReader();
        this.gestureDetector = new GestureDetector(sensor);

        this.sensor.IsAvailableChanged += Kinect_Status_Callback;

        bodyFrameReader.FrameArrived += this.Frame_Reader;

        sensor.Open();
    }
}

```

```

private void Kinect_Status_Callback(object sender, IsAvailableChangedEventArgs args)
{
    Console.WriteLine("Kinect Status: " + args.IsAvailable.ToString());
}

private void Frame_Reader(object sender, BodyFrameArrivedEventArgs args)
{
    bool datareceived = false; // check if any type of data has being received >>> Wait for a body appear
on screen
    int bodies_on_screen = 0;
    Body user_body = null; // user body - only body that will be used to compare with Gesture Library.
Will get Body.IsTracked from the Kinect if it is the only body on screen.

    using (BodyFrame bodyFrame = args.FrameReference.AcquireFrame())
    {
        if (bodyFrame != null)
        {
            if (this.bodies == null)
            {
                this.bodies = new Body[bodyFrame.BodyCount];
            }

            bodyFrame.GetAndRefreshBodyData(this.bodies); // Get All Body Data from bodyFrame and allocate
to the Body Vector
            datareceived = true;
        }
    }

    if (datareceived) // if BodyFrame is received call this function to deal with the new data
    {
        foreach (Body body in this.bodies)
        {
            if (body.IsTracked)
            {
                bodies_on_screen++;
                user_body = body;
            }
        }

        if (bodies_on_screen > 1 && bodies_on_screen != body_history)
        {
            Console.WriteLine("There are " + bodies_on_screen + " bodies captured");
            Console.WriteLine("This application only supports 1.");
            Console.WriteLine("Please Step Back.");
            body_history = bodies_on_screen;
            it_can_compare = false;
        }
        else if (bodies_on_screen == 1 && bodies_on_screen != body_history)
        {
            Console.WriteLine("1 Body is Tracked");
            Console.WriteLine("You are Ready to Go BABY!");
            body_history = bodies_on_screen;
            it_can_compare = true;
        }
        else if (bodies_on_screen < 1 && bodies_on_screen != body_history)
        {
            Console.WriteLine("There are " + bodies_on_screen + " bodies captured");
            Console.WriteLine("This application needs 1.");
            Console.WriteLine("Please Step In.");
            body_history = bodies_on_screen;
            it_can_compare = false;
        }

        if (it_can_compare)
        {
            Gesture_Compare(user_body);
        }
    }
}

private void Gesture_Compare(Body user_body) // look for user_body and compare with hand states and
gestures of the gesture library.
{
    //osc_message = new OSC_Messages(3, 1, 2, 3, 4); <<< Osc Prototype to remember how to use it.
    ulong trackingID = user_body.TrackingId;
    bool hand_data_received = false

```

```

//VISUAL BUILDER GESTURE COMPARE
if(gestureDetector.TrackingId != trackingID)
{
    gestureDetector.TrackingId = trackingID;

    if (trackingID == 0)
    {
        gestureDetector.IsPaused = true;
    }
    else
    {
        gestureDetector.IsPaused = false;
    }
}

//END VISUAL BUILDER GESTURE COMPARE

//RIGHT HAND COMPARE
if (user_body.HandRightState == right_hand_consistency && right_hand_consistency ==
HandState.Open)
{
    right_hand_score++;
    //Console.WriteLine(right_hand_score);

    if(right_hand_score > 90 && !right_cooldown) // 3 seconds of hand held = 90 frames of score
    (since kinect runs at 30 fps) <<< this can take more time if the machine doesn't read at smooth 30 fps from
    kinect
    {
        //Change Type HERE
        right_cooldown = true; // wait for a hand consistency drop to cooldown this movement

        if (type2_gate)
        {
            type2_gate = false; // close right hand gate (type 2)
            hand_data_received = true;
            //Console.WriteLine("Type 2 CLOSE");
        }
        else
        {
            type2_gate = true; // open right hand gate (type 2)
            hand_data_received = true;
            //Console.WriteLine("Type 2 OPEN");
        }
    }
}
else if (user_body.HandRightState != right_hand_consistency)
{
    right_hand_consistency = user_body.HandRightState;
    right_hand_score = 0; //resets "hold" time
    right_cooldown = false; // restore cooldown
}
//END RIGHT HAND

//LEFT HAND COMPARE
if (user_body.HandLeftState == left_hand_consistency && left_hand_consistency == HandState.Open)
{
    left_hand_score++;
    //Console.WriteLine(right_hand_score);

    if (left_hand_score > 90 && !left_cooldown) // 3 seconds of hand held = 90 frames of score
    (since kinect runs at 30 fps) <<< this can take more time if the machine doesn't read at smooth 30 fps from
    kinect
    {
        //Change Type HERE
        left_cooldown = true; // wait for a hand consistency drop to cooldown this movement

        if (type3_gate)
        {
            type3_gate = false; // close left hand gate (type 3)
            hand_data_received = true;
            //Console.WriteLine("Type 3 CLOSE");
        }
        else
        {
            type3_gate = true; // open left hand gate (type 3)
            hand_data_received = true;
            //Console.WriteLine("Type 3 OPEN");
        }
    }
}
else if (user_body.HandLeftState != left_hand_consistency)
{
    left_hand_consistency = user_body.HandLeftState;
}

```

```

left_hand_score = 0; //resets "hold" time
    left_cooldown = false; // restore cooldown
}
//END LEFT HAND

if (hand_data_received)
{
    gestureDetector.Database_Changer(type2_gate, type3_gate);
}

/*
if (osc_message != null)
{
    //osc_message.Send_OSC(); << Call in the end if osc message exist
}
*/
}
}
}

```

Figura 26: Código C# do módulo Translator, que perfaz a configuração dos parâmetros de comparação entre o banco de poses e os dados captados pelo Kinect

3.4. O módulo gerador e controlador musical

O módulo de geração e controle musical – aqui batizado de *Body Building Music Controller* (BBMC) – foi desenvolvido para possibilitar o controle de parâmetros musicais em tempo real através de informações enviadas via OSC. Seu desenvolvimento foi realizado como um *patch*²⁶ no software *Pure Data Vanilla* (PD), versão 0.48.1.

As mensagens de comando chegam através do protocolo de comunicação OSC até um decodificador. Após as mensagens serem recebidas, são distribuídas pelos três níveis, seguindo a ordem recebida pelo programa KG_OSC.

A mensagem de comando que chega ao nível 1 faz com este acione ou desligue um dos oito objetos sonoros contidos no repositório. O processador de efeito contido no nível 2 permanece inoperante até receber uma mensagem de acionamento (mão direita aberta). A partir de seu ligamento todos os objetos sonoros passam pelo processador de efeitos. Já os objetos sonoros passam pelo nível 3 continuamente, exceto quando o nível 2 estiver ligado, porém para que o processador dinâmico atue alterando os valores de

²⁶ Denominamos arquivos oriundos do programa Pure Data com a palavra inglesa *patch*.

volume da composição é necessário enviar uma mensagem de comando para sua entrada (mão esquerda aberta).

O fluxo de dados passa necessariamente pelo nível 1 para haja alguma forma de música, porém a utilização do nível 2 e 3 ficam a critério do atleta, podendo ou não serem utilizados durante a *performance*.

Figura 27: Fluxograma do gerador e controlador musical BBMC

O *patch* BBMC executa quatro funções: Recebe, decodifica e distribui as informações enviadas pelo KG_OSC; armazena e controla o disparo ou o desligamento

No primeiro bloco é configurado em ambos programas a porta de comunicação em que irão se comunicar. A porta de rede 9001 é a porta padrão de comunicação através de protocolo OSC do Pure Data.

O *patch* de comunicação mrpeach foi criado por Martin Peach e posteriormente desenvolvido pela comunidade online de programadores e desenvolvedores para PD. O *subpatch* foi responsável por receber as mensagens de acionamento de faixas musicais específicas, acionamento ou desligamento de níveis do sistema e recebimento de fluxo contínuo de dados para controle de parâmetros reguladores do filtro (nível 2) e de volume (nível 3).

O último bloco deste *subpatch* é responsável por distribuir as mensagens de acordo com a configuração pré-estabelecida entre as informações enviadas pelo aplicativo KG_OSC e os controles desenvolvidos no PD.

Figura 29: Parte do patch responsável pelo recebimento dos dados através do protocolo OSC

3.4.2 Máquina do Tempo (*sincronizador de disparo de objetos sonoros*)

Esta parte do controlador musical é responsável por sincronizar os disparos feitos pelo executor de objetos sonoros.

Dentre a realização da pose e o disparo do objeto sonoro ocorre um *delay* de aproximadamente 30 milissegundos. O atraso ocorre devido a programação de conferência entre a pose executada e o repositório de gestos pelo programa KG_OSC, que está programado para enviar a mensagem de comando apenas após identificar a realização da pose durante 30 frames seguidos.

Este *subpatch* é responsável para que um objeto sonoro não seja acionado fora do tempo estipulado para a peça musical.

O arquivo funciona como uma *gate* para a transmissão de dados, permitindo que só passe a informação de comando que aciona um objeto musical em uma janela de 250 milissegundos, o que acarretará, em uma peça composta no andamento de 120 BPM, o acionamento do objeto sonoro nos espaços de tempo relativos a 1/4 do tempo do total do compasso, o que se remete a figura da semínima, podendo um objeto sonoro ser acionado nos tempos fortes ou no contratempo dos outros objetos sonoros que já estão ativados.

Figura 30: *Subpatch* responsável pela sincronização dos objetos musicais.

3.4.3 Operador de nível 1: Executor de objetos sonoros

O operador de nível 1 é responsável pela leitura dos objetos musicais e seu disparo mediante informação de controle.

Ao iniciar o *patch* BBMC é necessário carregar todos os áudios contidos na pasta raiz de onde está o arquivo.

Foram configuradas oito saídas a partir de um roteador que separa as informações que chegam no nível 1. Cada saída está conectada a um *player* musical individual, com um objeto sonoro carregado em seu *buffer*. Cada executor tem duas saídas de áudios que enviam o som para o processador de efeitos ou direto para o *mixer* do *patch*.

Figura 31: Módulo responsável pelo controle de disparo dos objetos sonoros

3.4.4 Operador de nível 2: Processador de efeitos

O nível 2 recebe duas mensagens do programa KG_OSC, a primeira altera seu estado entre ligado e desligado²⁹. A segunda recebe informações de fluxo contínuo para controle da frequência central do filtro e do valor fixo para configurar o coeficiente de qualidade (Q).

A mensagem recebida, descompactada e enviada para o primeiro canal é o valor numérico de valor x, usado apenas como sinal para acionar um disparador, que por sua vez aciona um interruptor capaz de ligar o processador de efeito.

No segundo canal, a mensagem enviada pode variar entre 0 a 127³⁰. Dentro do *subpatch* há um conversor para transformar esses valores em uma escala em Hertz, com *range* entre 180 Hz e 12 KHz, valores possíveis de posicionamento da frequência central.

Para que atuação do efeito fosse controlada de maneira satisfatória adicionou-se uma função algorítmica no conversor de escala com o intuito de criar uma curva exponencial na forma de atuação do filtro. Este implemento faz com que o movimento do filtro nas frequências mais graves seja mais agressivo e na região aguda do espectro sua atuação seja mais lenta, tornando sua função artisticamente mais interessante para o sistema BBM.

²⁹ Os sinais de áudio que sofreram alteração do filtro terão a denominação de sinal *wet*. O sinal de áudio que não tiver interação com o processador de efeito chamaremos de *dry*.

³⁰ Foi adotado como padrão o número máximo de mensagens capaz de serem enviada via um canal do protocolo de comunicação MIDI.

Figura 32: *subpatch* responsável pelo efeito de filtro

3.4.5 Operador de Nível 3: Processador dinâmico

Este nível é responsável pelo controle do volume do sistema BBM. Semelhante ao nível 2, este nível também recebe duas informações, a de acionamento do *subpatch* e a de controle.

Recebendo valores escalonados de 0 a 127, o programa faz a conversão para escala de intensidade sonora demarcada em Bell, sendo possível a variação de -30 dB até 0dB SPL.

Foi adicionado ao controlador de dinâmica uma função algorítmica para se criar uma curva logarítmica, afim de que quando o atleta gesticule para controlar o processador dinâmico, haja maior atuação nos menores níveis de volume e menor atuação ao chegar perto de 0 dB. Além de artisticamente ser mais satisfatório essa função fornece um limiar de proteção para que não haja mudança abruptas em altos volumes.

Figura 33: Módulo responsável pelo controle de dinâmica

3.5. *Kinase*: uma composição modular interativa para BBM

Uma trilha musical foi composta pelo pesquisador do projeto para teste de validação do sistema BBM. Descreve-se a seguir o processo criativo da peça batizada de *Kinase*. Posteriormente, este processo será a matriz de um processo de autoria para que atletas-músicos possam criar performances coreográfica-musicais.

A composição foi estruturada segundo os conceitos de áudio dinâmico. Segundo Collins (2007) é considerado áudio dinâmico aquele que a estrutura reage a *inputs* externos.

A implementação em relação ao mapeamento de controles também levou em consideração a definição de Collins (2008), onde a composição se enquadra na categoria de música dinâmica controlada por gestos cinéticos. Segundo a autora, áudios não diegéticos, gerados a partir do controle físico do ator em alguma plataforma, são denominados como interação gestual cinética³¹

³¹ *Kinect gestural interaction* (COLLINS, 2008, p.127)

O processo de criação, programação de instrumentos virtuais, captação de instrumentos, mixagem e masterização se deram no programa Logic Pro X, versão 10.2.

Figura 34: Sessão da composição Kinase na DAW Logic Pro X

A instrumentação da peça conta com os instrumentos de percussão: bateria acústica, gongo e bateria eletrônica; cordas plectradas: guitarra; cordas friccionadas: quarteto de cordas; madeiras: flauta, oboé, clarinete e fagote; som sintetizado a partir de onda senoidal grave usado como linha de baixo; som sintetizado a partir de onda quadrada como linha melódica.

Com exceção das linhas de guitarras, que teve seu som discretizado a partir do instrumento real, todos os outros instrumentos foram emulados, tendo seus sons gerados a partir de repositórios digitais. Todos os instrumentos virtuais foram programados a partir do controlador Axiom 69, fabricado pela empresa M-Audio.

A bateria acústica utilizou o banco de som *sample based synthesis* nativo da DAW Logic Pro. O *preset* de sons usados foi a biblioteca Heavy+.

A bateria eletrônica utiliza samples pré-programados do sintetizador EXS 24, também nativo da DAW.

A guitarra, modelo stratocaster, utilizada na gravação foi captada com taxa de amostragem de 44.1 kHz de *sample rate* e 24 de *bit rate*. Para adição de efeitos foi utilizado o *plug in* Guitar Rig.

Os timbres usados na programação das cordas foram desenvolvidos pela empresa ProjectSam e está disponível na biblioteca Symphobia.

As madeiras utilizadas na composição desta peça estão disponíveis na livraria Claire Virtuoso, desenvolvida pela companhia 8Dio.

Os sons obtidos a partir da manipulação de osciladores digitais foram desenvolvidos no sintetizador EXS 24.

3.5.1. A Estrutura Musical de Kinase

A composição *Kinase*, por servir a uma proposta de controle e reprodução diferente do padrão usual, requereu algumas atenções em sua estruturação.

Figura 35: Estruturação da composição interativa

A ideia central da composição é desenvolvida levando em consideração a forma de controle do sistema BBM. Durante o uso do sistema, o atleta poderá acionar apenas um dos 8 objetos sonoros por vez, sendo assim, haverá inevitavelmente sobreposição de instrumentos harmônicos em diferentes momentos da composição, assim como a sobreposição dos instrumentos percussivos. A escolha da ordem de acionamento de cada

objeto sonoro, assim como o momento de acionamento farão soar um contexto musical diferente.

Um exemplo para entender o processo: imagine que o atleta faça a pose BDF, acionando o objeto sonoro Guitarra, após quatro compassos o atleta erige a pose AEC, disparando o objeto Lead/Synth. Enquanto o objeto sonoro está tocando o módulo 2 de sua estrutura, na região harmônica de lá menor, o objeto sonoro Lead/synth estará tocando o módulo 1 de estrutura, situado na região harmônica de Dm. Como cada linha escrita está programada em *lopping*, acaba ocorrendo muitas possibilidades diferentes de sobreposição dos módulos musicais, o que gerará distintas composições, baseadas nas escolhas de ordem e tempo de acionamento das poses pelo atleta.

Na composição tentou-se criar 3 climas distintos, porém que quando misturados, soassem bem. Há uma bateria acústica e uma linha de guitarra provindas do gênero *rock*, sintetizadores e *beat* eletrônico fundamentados na música eletrônica e instrumentos da orquestra clássica, construindo outras linhas. A partir das escolhas das poses, pode evidenciar apenas um estilo ou mistura-los

A composição foi estruturada em sete módulos, com quatro compasso em cada, e um módulo final, contendo dois compassos. Cada módulo contém e repete uma ideia musical diferente do próximo módulo. A partir do momento em que um objeto sonoro é disparado ele segue a sequência harmônica (figura 35): Dm9 (ré menor com sétima e nona), Am7 (lá menor com sétima), Dm9, C6 (dó maior com sexta), Esus (mi maior com quarta, sem terça), Am7, Em (mi menor) e Gsus (sol maior com quarta, sem terça)

Para cada instrumento a sequência melódica segue as tonalidades conforme indicadas em cada modulo da figura 35.

Para uma melhor compreensão, a partitura da peça Kinase está ao fim da dissertação no apêndice A.

Este processo composicional foi adotado devido as possibilidades de variação de arranjos orquestrais via controle gestual durante a execução. Note que o disparo de cada objeto sonoro pode ocorrer a qualquer momento, sobrepondo módulos que estão em tonalidades ou ritmos diferentes em suas sequências próprias, resultando numa variedade de movimentos harmônicos e rítmicos. Um outro exemplo que podemos imaginar é que houve o disparo das guitarras, após quatro compassos, o atleta disparou as cordas e, após 8 compassos o atleta dispara as madeiras. As guitarras estarão tocando o módulo

harmônico melódico (M h/m) 4 que se encontra no tom de C6 enquanto as cordas estarão executando o terceiro módulo, que se encontra na região de Dm 9 e as madeiras iniciarão o primeiro módulo (M h/m 1), também na região de Dm9. Esta sequência de poses escolhida pelo atleta acarretará na sobreposição de linhas melódicas e harmônicas que resultarão em uma quarta harmonia, diferente dos outros três materiais tocados separadamente.

A composição tem duração de 1 minuto, tempo médio de uma apresentação coreográfica da categoria de fisiculturismo masculino em competições realizadas pela IFBB³². Para que se houvesse uma divisão mais fácil em módulos de 4 compassos cada, escolheu-se a velocidade de 120 batidas por minuto, dando o total de 30 compassos quaternários.

O campo harmônico usado foi o de lá menor pentatônico, escolha dada para que houvesse a menor possibilidade da formação de acordes com muitas extensões dissonantes, tendo em vista a forma de controle do sistema BBM.

Na orquestração contamos com 5 instrumentos harmônicos (guitarra, quarteto de cordas, quarteto de madeiras, baixo sintetizado, sintetizador de onda quadrada) e 3 instrumentos percussivos (bateria acústica, bateria eletrônica e gongo). Cada linha de instrumento teve duas fases de construção:

- A primeira se deu no eixo horizontal sendo que cada instrumento tem uma linha melódica progressiva com movimentação independente do resto da orquestração. O processo criativo deste viés foi pensado de maneira seccionada, onde de 4 em 4 compassos há um movimento harmônico e melódico (M h/m), como pode ser visto na figura 35. Sendo assim temos 8 blocos com materiais melódicos e harmônicos diferentes para cada instrumento. Os instrumentos percussivos foram desenvolvidos utilizando a mesma linha de raciocínio.
- A segunda se deu no eixo vertical. Uma sequência harmônica, baseada no campo harmônico pentatônico de lá menor, foi escolhida para a estruturação musical. Porém devido a forma de funcionamento do sistema interativo BBM, onde só pode ocorrer o disparo de um objeto sonoro por vez, ocorre diferentes

³² Informações disponíveis no site oficial da federação IFBB de Mato Grosso do Sul. Disponível em: http://www.ifbbms.com.br/?page_id=65. Acessado em: 13 de maio de 2018.

sobreposições harmônicas, criando harmonias mais complexas do que a que foi usada como base inicial da composição.

Além de obedecer a estes requisitos estruturais, a composição foi criada levando em consideração a média dos gêneros musicais mais utilizados em campeonatos de fisiculturismo analisados com base na vivência dentro do esporte pelo pesquisador, a música eletrônica e o *rock*.

3.6 Sequencias de operações para o uso do sistema BBM

Para que seja possível o uso do sistema musical interativo controlado por performance coreográfica esportiva BBM, deve se seguir os seguintes passos:

1. Ligar o computador imbuído do sistema operacional Windows 10, com o Kinect *sensor* V2 conectado a ele através do Kinect *adapter*.
2. Conectar conversor analógico/digital para saída de áudio.
3. Ligar as caixas de som.
4. Dentro do sistema operacional, acionar o executável KG_OSC
5. Acionar o programa Pure Data, abrindo o patch BBMC
6. Se posicionar a frente do sensor e aguardar pela mensagem “*ready to go baby*” aparecer na tela de comando do programa KG_OSC

Após estes passos o sistema estará apto para uso.

4. Conclusões

Este capítulo apresenta e discute os resultados desta pesquisa, bem como tece considerações sobre trabalhos futuros.

4.1 Resultados

Ao fim deste projeto realizou-se a implementação de uma primeira versão do sistema musical interativo para uso em *performances* coreográficas esportivas, o *Body Building Music*, e uma *performance* registrada em áudio e vídeo, utilizando a peça interativa *Kinase*, também fruto desta elucubração.

A peça musical *Kinase*, foi interpretada com três variações em sua estrutura, afim de demonstrar algumas das possibilidades de concatenação dos materiais sonoros mediante a troca de poses e como a escolha das mesmas afetam a estrutura musical final.

Figura 36: *print screen* da peça audiovisual *Kinase*

Para melhor compreensão do sistema, seu funcionamento e registro da interpretação da obra *Kinase*, foram gravados sete vídeos e colocados no canal do pesquisador na

plataforma Youtube. Abaixo seguirá discriminado o nome e endereço virtual dos mesmos.

- Sistema Body Building Music – Controle de Nível 1 (Objetos Sonoros). Disponível em: < <https://youtu.be/7kLyCzJhTh8>>.
- Sistema Body Building Music – Controle de Nível 2 (Efeitos). Disponível em: <<https://youtu.be/usJ86zAxjMk>>.
- Sistema Body Building Music – Controle de Nível 3 (Volume). Disponível em: < <https://youtu.be/hrfHsJ291Sk>>.
- Sistema Body Building Music – *Middleware*. Disponível em: < <https://youtu.be/gvhpsoS-Va8>>.
- Kinase – Variação I. Disponível em: < https://youtu.be/V-HxNzMS_eg>.
- Kinase – Variação II. Disponível em: < <https://youtu.be/QaSuMxioaR0>>.
- Kinase – Variação III. Disponível em: < https://youtu.be/KMI7T3_XHdo>.

Através da realização desta pesquisa foi possível documentar um dos processos possíveis para desenvolver aplicações com uso do sensor Kinect versão 2.0, que utilizem o protocolo de comunicação OSC e interajam com o *freeware* Pure Data.

Enredar temas tão distintos através de uma tangente traz a possibilidade de levar temas não familiares até certos nichos culturais que de outra forma não teriam contato com a informação. Músico e compositores tem a oportunidade de entenderem a utilização da música no esporte fisiculturismo enquanto que atletas estreitam sua relação com parâmetros que definem uma composição musical, como estrutura, ritmo e arranjo instrumental, dando uma função social ao projeto que almeja lançar um olhar curioso à fronteira turva entre a arte inerente ao esporte e a *performance* esportiva existente na arte, incentivando a pesquisa entre estas áreas interdisciplinares.

Na seção teórica desta pesquisa fizemos um levantamento histórico da cultura *fitness* e como se deu sua relação com a música até chegar as apresentações coreográficas dos campeonatos de fisiculturismo.

O levantamento apresentado de sistemas que utilizam sensores de movimento e análise e comparação de suas estratégias de mapeamento, utilização de dispositivos e modo de operação, propiciou as bases do projeto para o sistema BBM e resultaram no artigo “O uso do Kinect como instrumento digital em sistemas interativos para

performance” apresentado no 15º Congresso de Engenharia de Áudio da AES (MONAZZI e FARIA, 2017).

O processo de implementação do sistema foi desafiador. A escolha do uso do Kinect 2.0 como dispositivo de sensoriamento do sistema acarretou uma série de especificidades, tanto de *hardware* quanto de *software*. Encontrar um microcomputador que atendesse aos requisitos de implementação para o sensor e que possibilitasse o uso total do potencial do equipamento, acabou se tornando mais intrincado do que o esperado, em razão dos requisitos impostos pelo sistema operacional proprietário da Microsoft. Tentativas de se utilizar plataformas portáteis microprocessadas – como o sistema Caninos Loucos desenvolvido na USP³³ – foram frustradas por não serem compatíveis com Windows 10.

Após configurar os equipamentos físicos necessários deu-se início a etapa de aprendizado e configuração dos *softwares* necessários para implementação do sistema. Ao todo foram necessário a configuração e uso de cinco programas diferentes para a montagem do sistema BBM: *Kinect Studio* para captação dos vídeos com metadados de coordenados de espaço; *Visual Gesture Builder* para marcação e treinamento dos gestos escolhidos para o banco de poses; *Microsoft Visual Studio* para forjar o aplicativo de leitura, decodificação, mapeamento e transmissão dos dados para o Pure Data; o gerador musical desenvolvido em PD e responsável pelo controle de disparo dos objetos sonoros, pelo acesso ao repositório de áudio e pela implementação dos processadores de efeito e dinâmica. Além destes, foi também utilizado a DAW *Logic Pro X* para composição, programação e mixagem da peça musical *Kinase*.

A utilização de tantos programas diferentes e com propostas de atuação em áreas distintas demandou boas horas de voo em cada ferramenta para realizar as implementações desejadas.

Todos os sistemas desenvolvidos estão disponíveis online para servirem como ponto de partida para outros desenvolvedores que forem trabalhar com algum dos programas aqui desenvolvidos em aplicações que necessitem de ajustes semelhantes. Tanto o executável criado no *Visual Studio* quanto o *patch* criado no PD podem servir para outros projetos que desejem trabalhar com mapeamento de gestos, sensor Kinect e protocolo de transmissão OSC.

³³ Programa Caninos Loucos, em <http://caninosloucos.org>. Acessado em 01/10/2018.

Ao final de todas as implementações no campo real e digital obtivemos um sistema capaz de reconhecer gestos do esporte fisiculturismo, captados através do sensor Kinect 2.0, e transforma-los em informações binárias para controlar um gerador musical.

4.2 Análise e discussão sobre os resultados

O caminho de intersecção entre música e esporte, mediado por aparato tecnológico, se mostrou ainda pouco explorado, sendo que achar bibliografia e sistemas que tenham aplicação similar se tornou um desafio para a construção do referencial teórico, o que culminou em decisões do próprio autor em relação a mapeamento, implementação e programação do sistema.

Os *softwares* que integraram o sistema BBM se mostraram eficazes, possibilitando sua montagem. Apesar de ter sido necessário o uso de diferentes *softwares* para a implementação, o que acarretou em um considerável tempo de aprendizado para uso das ferramentas, após o sistema estar configurado, o seu funcionamento se deu sem maiores problemas.

A implementação de um sistema de controle para ser usado durante uma *performance* coreográfica de fisiculturismo acarreta na necessidade de um sensor que não necessite de contato ou fique preso ao corpo do usuário. A utilização do Kinect para este fim se mostrou acertada, sendo o único tipo de sensor, com baixo custo de mercado, que demonstra atender as necessidades do sistema.

Dentre os testes realizados o sistema BBM exibiu desempenho satisfatório, atendendo aos objetivos iniciais da pesquisa.

O mapeamento entre gesto e objeto sonoro permitiu a criação e manipulação musical através do uso do sistema por qualquer pessoa que detivesse conhecimento das poses de fisiculturismo, sem ter a necessidade prévia de um conhecimento avançado na área musical.

O sensor Kinect mostrou ser eficaz neste tipo de implementação, tendo um bom desempenho na captação de imagem, porém variações de iluminação no ambiente e objetos/obstáculos no local podem afetar seu desempenho. Os testes mostraram que a eficácia do sensor diminui em ambiente com pouca iluminação ou com paredes e chãos

revestidos com materiais de alta reflexão, fazendo que o sensor não capte as poses ou aumentando a chance de haver erro na leitura de qual pose está sendo realizada.

Um dos problemas encontrados na implementação do sistema se deu por causa do repertório de poses que foi escolhido para ser mapeado. Como dentre as oito poses mapeadas há gestos similares, sendo que há dois pares de poses (DBF/DBC e EDF/EDC) cuja única diferença é sua realização de frente ou de costas, ocorreram casos de que a leitura de uma determinada pose ativou outro objeto sonoro (falsos positivos). As principais poses que apresentaram erros de mapeamento foram entre DBF e AEC, por causa do movimento da mão acima da cabeça, e PML e TML, devido a menor leitura de pontos pelo rastreamento do esqueleto por estas poses serem realizadas de lado para o sensor, permitindo apenas a leitura de um lado do corpo.

Para amenizar este problema foram gravados mais cliques destas poses e adicionados ao repositório das mesmas no VGB, para que a média estatística delas contasse com mais dados no momento de sua compilação. Também foram adicionadas poses com mapeamento negativo no VGB dentro do repositório das poses que estavam sendo confundidas. Assim, no repositório da pose DBF foi inserido um clipe com a pose AEC e vice-versa, com marcação negativa para auxílio na diferenciação das poses. A mesma estratégia foi adotada nas poses PML e TML.

Outro ponto reportado se dá devido ao atraso entre a realização da pose e o disparo do objeto musical. Este *delay* ocorre pelo limiar (*gate*) inserido que configura a necessidade do reconhecimento da pose durante 10 *frames* para que seja disparada a mensagem de comando.

Pontos que devem ser abordados para melhorar o desempenho do sistema incluem a gravação de mais cliques das poses por pessoas diferentes em ambientes diferentes, o que auxiliará numa melhor marcação dos pontos a serem analisados pelo módulo de *machine learning* do VGB e melhorará o reconhecimento das poses, bem como um maior número de testes da aplicação por atletas, para ajuste fino de parâmetros artísticos do sistema.

A composição musical Kinase serviu adequadamente para os fins propostos pelo sistema. O principal desafio da composição dinâmica é se tornar interessante auditivamente quando manipulada mediante as limitações de controle do sistema. A partir do disparo individual dos objetos sonoros que compõe a peça e a possibilidade de um grande número de sobreposições de elementos diferentes, cria-se uma grande quantidade

de resoluções musicais da peça, diferentes entre si. Resultado que agrega ao sistema, possibilitando mais escolhas de interação entre atleta e o BBM.

A fim de se colocar à prova o sistema interativo e sua aplicação, chamamos a atleta profissional Diana Monteiro³⁴, para testá-lo e relatar sua primeira impressão e uma avaliação acerca da eficácia do sistema de criação/geração musical desenvolvido foi feita por meio de uma entrevista³⁵.

Em seu depoimento a atleta relatou que acredita que o sistema pode vir ajudar atletas a melhorarem a qualidade de suas poses através do sistema uma vez que só há o disparo de um objeto sonoro mediante a realização da pose de forma correta. Trabalhar com o sistema a fim de se trabalhar a rotina de poses para um campeonato também se torna válido.

Ressaltando a carência de pesquisas com foco no fisiculturismo, Diana acredita que o trabalho leva informações sobre o esporte para novas áreas, ajudando a públicos distintos terem contato e compreenderem sobre esta categoria esportiva que ainda hoje sofre discriminação por alguns grupos sociais.

Diana ainda vê a possibilidade de aplicação do sistema em esportes praticados por pessoas com deficiência visual. A atleta acredita que o mapeamento de determinados movimentos pelo sistema para que haja uma validação sonora quando realizados corretamente poderia ser de grande utilidade para os esportes praticados por portadores de necessidades especiais.

Futuramente, poderão ser definidos e explorados, para tanto, critérios e parâmetros para medição de eficácia e de eficiência do sistema de criação/geração musical, que poderão abordar aspectos acerca da validade e relevância composicional ou de *performance* em música, bem como aspectos de eficácia do mapeamento proposto (por

³⁴ Diana Monteiro é graduada em Educação Física (Centro Universitário Moura Lacerda) e Nutrição (UNIP-RP). Especialista em Fisiologia do Exercício Universidade Veiga de Almeida) e Treinamento Desportivo (UNIGRANRIO). Mestranda no programa de Pós-Graduação da Escola de Educação Física e Esporte da USP-RP. Atleta profissional pela confederação IFBB pro Elite, é Tetracampeã Mundial de *Fitness* e três vezes participante da competição Mr. Olympia na categoria *fitness* coreográfico.

³⁵ Entrevista realizada no LATM (Laboratório de Acústica e Tecnologia Musical), situado no Departamento de Música da Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto - SP. Pode ser acessada online, na plataforma YouTube, no canal pessoal do pesquisador. Disponível em: <https://youtu.be/RDYDa62pDaQ>.

exemplo, comparando-se performances diferentes), e aspectos de resolução e controlabilidade (desempenho técnico) do sistema proposto.

4.3 Considerações Finais

Através do processo de pesquisa e documentação realizados para a concatenação deste sistema obteve-se a elucidação de pontos necessários para implementações interativas que desejem trabalhar com as ferramentas aqui discutidas.

O ponto que diferencia o BBM é a sua área de aplicação. A forma de fazer música através de um sistema musical digital cria uma possibilidade de abordagem diferente da relação entre música e esporte já existente, provocando atletas a compreenderem gestuais musicais e músicos entenderem o gestual esportivo para se poder chegar a um novo ponto de diálogo.

A utilização do sistema como forma de expressão artística/esportiva resultou em um ambiente propício para intercâmbio de ideias e ebulição de novos projetos, possibilitando o atleta a explorar competências de ordem musical, e ao músico vivenciar os gestuais desportivos, experienciando suas possíveis relações causais com os gestos musicais.

Podemos observar, através de teste por pessoas leigas, tanto em música quanto nas regras de poses do fisiculturismo, que o sistema ao ser controlado através de gestual físico, tem uma curva de aprendizado menor em relação ao tempo requerido para o domínio da forma de uso de um instrumento tradicional, possibilitando sua utilização por um maior número de pessoas.

Podemos assim evidenciar cinco heranças, descritas nos próximos parágrafos, que a realização do sistema BBM deixa e que podem ser úteis em trabalhos futuros.

O uso do sistema interativo pode ter caráter didático. Conforme conversas com profissionais do fisiculturismo, o sistema poderia ser usado para o treino de poses, já que as faixas musicais só são disparadas mediante a realização da pose de forma que esteja em concordância com as regras da federação internacional de *bodybuilding*.

Do ponto de vista social, o sistema interativo pode auxiliar a atletas com necessidades especiais. Um exemplo é sua aplicação junto a deficientes visuais com o objetivo de treinarem determinado gesto ou movimento. A partir do momento que é possível criar

uma ferramenta de reconhecimento gestual, um banco de dados com gestos previamente gravados, e há a possibilidade de se alinhar estímulos sonoros a este sistema, temos uma ferramenta com boa aplicabilidade para treinamentos físicos associados ou sincronizados a uma estruturação musical.

O processo para se compor a peça musical dinâmica aqui documentado é inédito. O autor não encontrou bibliografia de procedimento similar e aplicação semelhante até o presente momento. Para criar um projeto de controle musical por coreografia esportiva um atleta-músico terá de seguir uma sequência de operações para autoria semelhante à descrita no item 3.5. Para tanto o atleta deve definir o mapeamento desejado entre as poses disponíveis no banco de poses construído e os objetos sonoros.

O processo criativo da composição pode ser utilizado por outros sistemas audiovisuais com o intuito de se criar músicas diferentes com a mesma matéria prima, apenas alternando a instrumentação e os pontos de partida de cada faixa. Um exemplo de implementação é na área de composição de música para vídeo games, onde pode-se trocar o controle exercido pelo atleta pelas decisões de movimento do avatar digital feita pelo jogador.

Artisticamente o sistema desempenha o papel de objeto indagador, despertando o estranhamento à primeira vista das junções propostas, e em um segundo momento, suscitando a curiosidade e o interesse pelos temas transpassados.

REFERÊNCIAS BIBLIOGRÁFICAS

BARGSTEN, Joey. **Gesture and Performance with Kinect, Quartz Composer and Pure Data**. Florida Atlantic University, 2014.

BELLONA, Jon. **Kinect Via Synapse**. Artigo disponível em <<https://github.com/jpbellona/Kinect-Via-Synapse>>. Acessado em 10 de fevereiro de 2016.

BERGER, Kai; MEISTER, Stephan; NAIR, Rahul; KONDERMANN, Daniel. **A State-of-the-Art Report on Kinect Sensor Setups in Computer Vision**. Livro *Time of Flight and Depth Imagins*, p. 257-272. Springer-Verlag. Berlin, 2013.

BERGLAND, Andreas; WECHSLER, Robert. **Composing Interactive Dance Pieces for the MotionComposer, a device for Persons with Disabilities**. New Interfaces for Musical Expression, Baton Rouge, LA, USA, 2015.

BOURDIEU, Pierre. **Como é Possível Ser Esportivo?** Questões de sociologia, Rio de Janeiro: Marco Zero; 1983. P.136-53.

BROWNE, Murray. **Interactive Music: Balancing creative freedom with musical development**. Tese de doutorado. Queen Mary University of London. Londres, 2012

CARDOSO, Gabriel S. **Microsoft Kinect: Criando aplicações interativas com o Microsoft Kinect**. Casa do Código. 2013.

COLLINS, Karen. **An Introduction to the Participatory and Non-Linear Aspects of Video Game Audio**. Helsinki, 2007.

_____. **Game Sound: an introduction to the history, theory, and practice of video game music and sound design**. MIT Press, Cambridge, 2008.

DARÉ, Paula Serafim. **O corpo na obra de arte: Idade Média, Renascimento e Maneirismo**. 2015. Dissertação – Instituto Junguiano de São Paulo. São Paulo.

DVALIDZE, Irina. **Iris: A Beautiful New Take on XBOX KINECT**. Disponível em: <<http://animalnewyork.com/2012/iris-kickass-new-take-on-microsoft-kinec/>>. Acessado em: 8 de outubro de 2015

FARNELL, Andy James. **Synthetic game audio with Puredata (Summary)**. Disponível em: <<http://obiwannabe.co.uk/html/papers/audiomostly/handout.pdf>>. Acessado em 9 de setembro de 2015

FORNARI, Jose. **Designing Bodiless Musical Instruments**. Proceedings of AES Brasil. 8 –10 maio 2012. Expo Center Norte. São Paulo. Brasil. 2012.

GNECCO, Bruno B DIAS, Diego R. C., BRASIL, Gustavo J. C., GUIMARÃES, Marcelo P. Marcelo P. **Desenvolvimento de Interfaces Naturais de Interação usando o Hardware Kinect**. Corollarium Tecnologia, 2012.

GUMBRECHT, Hans Ulrich. **Elogio da Beleza Atlética**. Companhia das Letras. São Paulo, 2007.

HENRIQUE, Fábio. **Guia de Mixagem 2**. Rio de Janeiro: Música & Tecnologia, 2008.

_____. **Guia de Mixagem**. 2ª edição. Rio de Janeiro: Música & Tecnologia, 2007.

HJELMSLEV, Louis. **Prolegômenos a uma Teoria da Linguagem**. Perspectiva. Brasil, 2009.

HUNT, Andy; WANDERLEY, Marcelo M.; KIRK, Ross. **Towards a Model for Instrumental Mapping in Expert Musical Interaction**. IRCAM. Paris, France, 2000. Disponível em: <http://recherche.ircam.fr/anasy/wanderle/Gestes/Externe/Hunt_Towards.pdf>. Acesso em: 20 de julho, 2017.

IAZETTA, Fernando. **Música e Mediação tecnológica**. Brasil. Editora Perspectiva, 2009 Images, Animation, and Interaction. Morgan Kaufmann Publishers is an imprint of Elsevier,

_____. “Interação, Interfaces e Instrumentos em Música Eletroacústica,” in *Proceedings of the II ‘IHC - Interação Humano-Computador’ Conference* Campinas: Unicamp, 1998

IFBB. PANAMERICAN GAMES EVENT 15 AUGUST 2018. Disponível em: <<https://ifbb.com/event/panamerican-games/>>. Acesso em: 20 set. 2018.

JENSENIUS, Alexander Refsum. **Kinectofon: Performing with Shapes in Planes**. In: International Conference on New Interfaces for Musical Expression, 27 de Maio, 2013, Daejeon. Anais eletrônicos, Daejeon: NIME, 2013. Disponível em: <http://nime.org/proceedings/2013/nime2013_220.pdf>. Acesso em: 23 de julho de 2017

JORDÃO, Alexandre de Medeiros. **Cinema e o Culto ao Corpo: Rambo e Capitão América como Personal Trainer**. 2013. Dissertação (Mestrado em Comunicação e Cultura). Programa de Pós-Graduação em Comunicação e Cultura, Universidade de Sorocaba.

KARAGEORGHIS, Costas, PRIEST, David-Lee. **Music in Sport and Exercise: An Update on Research and Application**. The Sport Journal. Disponível em: <<https://www.questia.com/library/journal/1G1-210521313/music-in-sport-and-exercise-an-update-on-research>>. Acessado em: 10 de setembro de 2015

KARONGA, Austin. **Bodybuilding for Tokyo Olympics**. Daily News journal. 17 Nov. 2016. Disponível em: < <https://www.dailynews.co.zw/articles/2016/11/17/bodybuilding-for-tokyo-olympics>>. Acesso em: 20 jun. 2018.

KRONLACHNER, Matthias. **The Kinect distance sensor as human-machine-interface in audio-visual art projects**. Project Report for the Institute of Electronic Music and Acoustics. Graz, 2013.

LABAN, Rudolf von; ULLMANN Lisa. **Choreutics**. Macdonald & Evans. Londres, 1966

LACHAT, Elise; LANDES, Tania; MACHER Hélène; GRUSSENMEYER, Pierre. **Assessment and Calibration of a RGB-D Camera (Kinect v2 Sensor) Towards a Potencial Use for Close-Range 3D Modeling**. Journal Remote Sensing. 2015, 7, p.13070-13097. Strasbourg, 2015. Disponível em: <https://www.mdpi.com/2072-4292/7/10/13070>. Acesso em: 15 de julho de 2018.

LIMA, Ricardo Rodrigues de. **“Música e(m) Cena: processo de criação em mídias diversas”**. Dissertação de Mestrado, PUC, Programa de Pós-Graduação em Comunicação e Semiótica, São Paulo. 2006

LOCKS, Adam; RICHARDSON, Niall. **Critical Readings in Bodybuilding**. Routledge. Nova York. 2012.

MAGLIONE, Júlio. IFBB meeting on the next Pan American Games in 2019. Acessado em: 20 de maio de 2016. Disponível em: <<http://www.ifbb.com/2016/04/ifbb-meeting-on-the-next-panamerican-games-in-2019>>

MATESCO, Viviane. **Corpo, ação e imagem: consolidação da performance como questão**. Revista Poiésis, n. 20, p. 105-118. Rio de Janeiro, 2012.

MCGOWAN, Jim. **Kinectori-On – A Kinect Matrix Sequencer**. Hong Kong, 2012. Disponível em: <<http://bleepsandpops.com/post/25845714139/kinectori-on-a-kinect-matrix-sequencer>> Acessado em: 12 de setembro de 2015

MENEGUETTE, Lucas. **Áudio Dinâmico para Games: Conceitos Fundamentais e procedimentos de Composições Adaptativa**. Brasil. Proceedings of SBGames. 2011

MIRANDA, Eduardo R. E WANDERLEY, Marcelo. **New Digital Instrument: control and interaction beyond keyboard**. Middletown: A-R Editions, 2006.

MISKALO, Vitor Kisil. **A performance enquanto elemento composicional na música eletroacústica interativa**. Dissertação de Mestrado, USP, Programa de Pós-Graduação em Música, São Paulo 2009

_____. **¿Música?: processos e práticas de criação e performance em um ambiente de pesquisas em sonologia**. Dissertação de Doutorado, USP, Programa de Pós-Graduação em Música, São Paulo, 2014.

MONNAZZI, João P. S.; FARIA, Regis R. A. **O uso do Kinect como instrumento digital em sistemas interativos para performance.** In: 15º Congresso de Engenharia de Áudio, 2017, Florianópolis. Anais do 15º Congresso de Engenharia de Áudio da AES Brasil. Florianópolis: Universidade Federal de Santa Catarina, 2017. p. 33-40. Disponível em <<http://www.aesbrasil.org/congressos/anais/>>.

MONTANHA, Fausto Amaro R. P.; HELAL, Ronaldo George. **Corpo, performance e materialidade: por um olhar não-hermenêutico nos estudos sobre esporte.** Fronteiras- estudos midiáticos. Rio de Janeiro, 2013.

MULDER, Axel G. E. *Towards a choice of gestural constraints of instrumental performers.* In: *Trends in Gestural Control of Music*. Paris: Ircam, p. 315-335, 2000.

NEVES, Thiago Gonçalves. **Os Primórdios do Halterofilismo e Fisiculturismo no Brasil.** Porto Alegre, 2012.

NICOLAI, Klaus; GEELHAAR, Jens. **Technical specifications of MotionComposer.** Disponível em:< <http://internal.motioncomposer.com/technical>>. Acessado: 13 de maio de 2017.

ORIGEM DA PALAVRA. Disponível em: <http://origemdapalavra.com.br/pergunta/fisiculturismo/>. Acessado em: 05 de junho de 2018.

PADOVANI, José Henrique, FREIRE, Sérgio. **Explorando envoltórias espectrais em sistemas musicais interativos.** XVI Congresso da Associação de Pesquisa e Pós-graduação em Música (ANPPOM). Brasília, 2006

PAGLIARI, Diana; PINTO, Livio. **Calibration of Kinect for Xbox One and Comparison between the Two Generations of Microsoft Sensors.** Journal Sensor.2015, 15, p.27569-27589. Milão, 2015. Disponível em:< <https://www.mdpi.com/1424-8220/15/11/27569/htm>>. Acesso em: 15 de julho de 2018.

PALMIERI, Alan. **Power of Mind.** 2005. Disponível em: <<http://www.palmieribodybuilding.com/Power%20of%20The%20Mind.pdf>>. Acessado em: 24 de julho de 2016.

PATRÍCIO, Eduardo Luís Brito. **Instrumentos Musicais Digitais.** 2010. Dissertação (Mestrado em Música) – Faculdade de Música, Universidade Federal do Paraná, Paraná, 2010.

PAULA, Felipe José Teixeira de, MOREIRA, William de Carvalho, CARVALHO Marcos Alberto de. *Interface Natural Utilizando o Microsoft Kinect.* Revista Eletrônica de Ciência da Computação v.9 n.1 UNIFENAS, 2014.

PHELAN, Peggy. **Unmarked: The Politics of Performance**. Routledge. Nova York, 1993.

PIERIN, André. [comentário pessoal]. **Facebook**. 18 de julho de 2016. Disponível em: <<https://www.facebook.com/andre.paje>>. Acessado em: 23 de julho de 2016

RAHMAN, Mansib. **Beginning Microsoft Kinect for Windows SDK 2.0**. Apres. Quebec, 2017.

ROKEBY, D. Disponível em: <<http://www.davidrokeby.com/vns.html>>. Acessado em: 20 de março de 2016.

ROSS, Alex. **Escuta Só**. Brasil. Companhia das Letras, 2010

_____. **O Resto é Ruído**. Brasil. Companhia das Letras, 2011

ROVAN, Joseph WANDERLEY, Marcelo; DUBNOV, Shlomo; DEPALLE, Philip. **Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance**. In: Proceedings of the AIMI International Workshop, Genova, Associazione di Informatica Musicale Italiana, p. 68-73, 1997.

ROWE, Robert. **Interactive Music Systems**. MIT Press, 1993.

SALTER, Cris. **Entangled. Technology and the Transformation of Performance**. MIT Press. Massachusetts. 2010.

SANTOJA, Rafael. **The History of the IFBB form 1946 till the present time**. 2016. Disponível em: www.ifbb.com. Acessado em: 15 de junho de 2018.

SAS INSTITUTE. **Machine Learning: o que é e qual é sua importância**. 2018. Disponível em: <https://www.sas.com/pt_br/insights/analytics/machine-learning.html>. Acessado em: 23 de maio de 2018.

SCHECNER, Richard. **What is performance?** Performance Studies: An Introduction, second edition. New York e Londres, 2006.

SCHLEGEL, homepage. Biblioteca OSC P5 desenvolvida para Processing. Disponível em: <<http://www.sojamo.de/libraries/oscP5/>>. Acessado em 2 de outubro de 2015.

SCHWARZENEGGER, Arnold; HALL, Douglas Kent. **Arnold: The Education of a Bodybuilder**. Simon e Schuster. Nova York, 1977.

SCHWARZENEGGER, Arnold. **The New Encyclopedia of Modern Bodybuilder**. Simon e Schuster. Nova York, 1998.

SENTURK, Sertan *et al.* Crosssole: A Gestural Interface for Composition, Improvisation and Performance using Kinect. Disponível em:

<http://www.nime.org/proceedings/2012/nime2012_185.pdf> Acessado em 10 de setembro de 2015

SHIFFMAN, Daniel. Código do exemplo 16-13: Simple Motion Detection, do livro *Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction*. Disponível em: <<http://www.learningprocessing.com/examples/chapter-16/example-16-13/>>. Acessado em 8 de outubro de 2015.

_____. **Getting Started with Kinect and Processing**. Disponível em: <http://shiffman.net/p5/kinect/>. Acessado em: 10 de outubro de 2015

_____. **Learning Processing: A Beginner's Guide to Programming**. ELSEVIER INC. Burlington, 2008

SILVA, Jaziel Vitalino Souza, CALEGARIO, Filipe, GIORDANO Cabral, RAMALHO Geber. **Avaliando Interfaces Gestuais para Prática de Instrumentos Virtuais de Percussão**. 2013. 12 p. Artigo – Centro de Informática – Universidade Federal de Pernambuco. 2013.

SILVA, Jaqueline Rigon da; GRESS, Flademir Ari Galvão. **A Influência da Música e Ritmos Musicais nos Exercícios Físicos Resistidos**. Ver. Acta Brasileira do Movimento Humano – Vol.2, n.4, p.46-59 – Out/Dez, 2012.

SILVA, Marcos. **Resenha de Elogio da Beleza Atlética de Hans Ulrich Gumbrecht**. Companhia das Letras. 2007.

SILVEIRA, Fabrício. **Resenha Além da atribuição do sentido**. São Leopoldo, 2010.

STEINER, Hans-Christoph. bang | pure data, **chapter Building your own instrument with Pd, Wolke Verlag, 2006**. Artigo disponível em <http://puredata.info/groups/pdgraz/label/book01/bang_09_steiner_72-79.pdf>. Acessado em 17 de setembro de 2015.

SWEET, Michael. **Writing Interactive Music for Video Games**. Pearson Education Crawfordsville, 2015.

TAVARES, Tiago Fernandes et al. **Cooperative musical creation using Kinect, WiiMote, Epoc and microphones: a case study with MinDSoundS**. IEEE International Conference on Systems, Man, and Cybernetics. 2015.

TUBINO, Manoel José Gomes. **Dicionário enciclopédico Tubino de esportes**. Samus. São Paulo, 2007.

WALLACE, David Foster. David Foster Wallace on Unwordly Athletes. **ESPN.com**. 20 de outubro de 2009. Disponível em: <http://www.espn.com/blog/truehoop/post/_/id/5383/david-foster-wallace-on-unworldly-athletes>. Acesso em: 20 de out. 2018.

WECHSLER, Robert; BERGSLAND, Andreas. **Jeu de Modes**. Hexagram Black Box, Concordia University, Montreal. Canada, 2014. Disponível em:<<http://www.palindrome.de/infos/jeu.htm>>. Acessado: 13 de maio de 2017

WIJNALDA, Gertjan L. **A personalised music system for motivation in sport performance**. Tese de Mestrado apresentada para VRIJE Universiteit. Amsterdam, 2005.

WINKLER, Todd. **Making Motion Musical: Mapping Strategies for Interactive Computer Music**. Publicado em *Proceedings of the 1995 International Computer Music Conference*, 1995.

**APÊNDICE A – Partitura de Kinase: composição dinâmica para sistema musical
interativo**

Kinase

BPM 120

Dm⁹

Jom Monnazzi

Flute *stacatto...*

Oboe

Clarinet in B $\flat$

Bassoon

Drum Eletronic

Drum Acoustic

Electric Guitar *overdrive on*

Synth Bass

Synth/Lead (Sawtooth)

Violin I *mp*

Violin II *mp*

Viola *mp*

Basso + Cello *mp*

f

2 Am^7

5

Fl.

Ob.

Cl.

Bsn.

Dr. El

Dr. Ac.

E. Gtr.

S. Bass

Lead 2

Vln. I

Vln. II

Vla.

B + Vc.

Dm⁹ 3

9

Fl. *divise...*

Ob.

Cl.

Bsn.

Dr. El

Dr. Ac.

E. Gtr.

S. Bass

Lead 2

Dm⁹

Vln. I

Vln. II

Vla.

B + Vc.

4 C^6

13

Fl.

Ob.

Cl.

Bsn.

Dr. El

Dr. Ac.

E. Gtr.

S. Bass

Lead 2

Vln. I

Vln. II

Vla.

B + Vc.

13

E(sus4)

16

Fl.

Ob.

Cl.

Bsn.

Dr. El

Dr. Ac.

E. Gtr.

S. Bass

Lead 2

Vln. I

Vln. II

Vla.

B + Vc.

divise

legato

legato

legato

legato

marcato

marcato

marcato

marcato

122

6

Am

19

Fl.

Ob.

Cl.

Bsn.

Dr. El

Dr. Ac.

E. Gtr.

S. Bass

Lead 2

Vln. I

Vln. II

Vla.

B + Vc.

Em

7

22

Fl.

Ob.

Cl.

Bsn.

Dr. El

Dr. Ac.

E. Gtr.

S. Bass

Lead 2

Vln. I

Vln. II

Vla.

B + Vc.

divise

Detailed description: This musical score page contains measures 22 through 25. The woodwind section (Flute, Oboe, Clarinet, Bassoon) plays sustained notes with long slurs. The percussion section includes Electric Drums (Dr. El) and Acoustic Drums (Dr. Ac.). The electric guitar (E. Gtr.) plays a rhythmic pattern of eighth notes, while the string bass (S. Bass) plays a steady eighth-note line. A lead instrument (Lead 2) plays a melodic line with eighth notes. The string section consists of Violin I (Vln. I), Violin II (Vln. II), Viola (Vla.), and Bass/Viola (B + Vc.), all playing eighth-note patterns. The Viola part includes the instruction 'divise' at the end of measure 25. The key signature is one flat (E-flat major or D minor), and the time signature is 4/4.

8

26

Fl.

Ob.

Cl.

Bsn.

Dr. El

Dr. Ac.

E. Gtr.

S. Bass

Lead 2

Vln. I

Vln. II

Vla.

B + Vc.

divise

divise

The musical score for measures 26-28 is as follows:

- Flute (Fl.):** Measure 26 has a whole rest. Measures 27 and 28 have whole rests.
- Oboe (Ob.):** Measure 26 has a whole rest. Measures 27 and 28 have a half note G4, marked *divise*.
- Clarinet (Cl.):** Measure 26 has a whole rest. Measures 27 and 28 have a half note G4, marked *divise*.
- Bassoon (Bsn.):** Measure 26 has a whole rest. Measures 27 and 28 have a half note G4.
- Drums (Dr. El, Dr. Ac.):** Measure 26 has a drum pattern of eighth notes. Measure 27 has a drum pattern of eighth notes. Measure 28 has a drum pattern of eighth notes.
- Electric Guitar (E. Gtr.):** Measure 26 has a chord of G4 and B4. Measure 27 has a chord of G4 and B4. Measure 28 has a chord of G4 and B4.
- String Bass (S. Bass):** Measure 26 has a quarter note G2. Measure 27 has a quarter note G2. Measure 28 has a quarter note G2.
- Lead 2:** Measure 26 has a quarter note G4. Measure 27 has a quarter note G4. Measure 28 has a quarter note G4.
- Violin I (Vln. I):** Measure 26 has a quarter note G4. Measure 27 has a quarter note G4. Measure 28 has a quarter note G4.
- Violin II (Vln. II):** Measure 26 has a quarter note G4. Measure 27 has a quarter note G4. Measure 28 has a quarter note G4.
- Viola (Vla.):** Measure 26 has a quarter note G4. Measure 27 has a quarter note G4. Measure 28 has a quarter note G4.
- Bass/Violoncello (B + Vc.):** Measure 26 has a quarter note G2. Measure 27 has a quarter note G2. Measure 28 has a quarter note G2.

29 **G(sus4)** 9

Fl.

Ob.

Cl.

Bsn.

Dr. El

Dr. Ac.

E. Gtr.

S. Bass

Lead 2

Vln. I

Vln. II

Vla.

B + Vc.

**ANEXO A – *Datasheet* do requerimento da configuração necessário para utilizar o
Kinect sensor V2**

03/10/2018

System Requirements | Microsoft Docs

System Requirements

10/21/2014 2 minutes to read

You must meet the following system requirements to develop applications using the Kinect for Windows SDK version 2.0.

ⓘ Important

Refer to the release notes for the most up-to-date info about system requirements and installation instructions.

- Supported Operating Systems and Architectures
- Recommended Hardware Configuration
- Software Requirements
- Additional Software Requirements

Supported Operating Systems and Architectures

The following operating systems and architectures are supported:

- Windows 8 (x64)
- Windows 8.1 (x64)
- Windows 8 Embedded Standard (x64)
- Windows 8.1 Embedded Standard (x64)

Recommended Hardware Configuration

Your computer must have the following minimum capabilities:

- 64 bit (x64) processor
- 4 GB Memory (or more)
- I7 3.1 GHz (or higher)
- Built-in USB 3.0 host controller (Intel or Renesas chipset).

If you're adding USB 3.0 functionality to your existing PC through an adapter, please ensure that it is a Windows 8 compliant device and that it supports Gen-2. See the troubleshooting section of [Getting Started](#) for more information.

03/10/2018

System Requirements | Microsoft Docs

- DX11 capable graphics adapter (see list of known good adapters below)
 - Intel HD 4400 integrated display adapter
 - ATI Radeon HD 5400 series
 - ATI Radeon HD 6570
 - ATI Radeon HD 7800 (256-bit GDDR5 2GB/1000Mhz)
 - NVidia Quadro 600
 - NVidia GeForce GT 640
 - NVidia GeForce GTX 660
 - NVidia Quadro K1000M
- A Kinect v2 sensor, which includes a power hub and USB cabling.

① Important

Many laptops switch to a low-power mode that reduces GPU performance when operating without external power. If you plan to work on a laptop that is operating on battery power, you should disable low-power mode for your GPU. In some cases, the GPU performance is limited while on battery power even when low-power mode is disabled. In this case, you should keep your laptop plugged in while using the Kinect for Windows sensor.

① Important

Kinect Fusion - When using Kinect Fusion, Kinect can process data either on a DirectX 11 compatible GPU with C++ AMP, or on the CPU, by setting the reconstruction processor type during reconstruction volume creation. The CPU processor is best suited to offline processing as only modern DirectX 11 GPUs will enable real-time and interactive frame rates during reconstruction.

Software Requirements

The following developer environments are supported:

- All Visual Studio 2012, including Visual Studio 2012 Express ([Microsoft Visual Studio 2012 Express](#))
- Visual Studio 2013 Ultimate, Premium, Professional, and Express for Windows Desktop

Additional Software Requirements

The speech samples also require:

- [Microsoft Speech Platform Software Development Kit \(Version 11\)](#), if you need to compile the samples.
- [Microsoft Speech Platform Runtime \(Version 11\)](#), which is automatically installed as part of the Kinect runtime setup, if you want to run the samples only.

The Depth-D3D and DepthWithColor-D3D samples also require:

[https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn782036\(v=ie8.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn782036(v=ie8.10))

03/10/2018

System Requirements | Microsoft Docs

- [DirectX Software Development Kit](#), if you need to compile the samples.
- [DirectX End-User Runtimes \(June 2010\)](#) if you want to run the samples only.

**ANEXO B – Diretrizes normativas das poses mandatórias para categoria
fisiculturismo masculino**

Confederación Sudamericana de Fisico Culturismo y Fitness

Reconocida por la IFBB Internacional - 2006

Argentina • Aruba • Bolivia • Brasil • Chile • Colombia • Ecuador • Paraguay • Peru • Uruguay • Venezuela

MODIFICACIONES EN EL CULTURISMO MASCULINO

POSES COMPARATIVAS

Tomando en cuenta la propuesta del Comité de Jueces encabezado por el Sr. Pawel Filleborn, el Consejo Ejecutivo de la IFBB ha aprobado las modificaciones solicitadas en las poses comparativas que se realiza en el físico culturismo masculino y culturismo clásico. Las mismas que entraran en vigencia desde este año (2015).

Las modificaciones son mínimas, se busca que los atletas realicen las poses obligatorias de manera homogénea, brindando la posibilidad de que los jueces aprecien de mejor manera la estructura muscular individual y en conjunto. Analicemos como se plantean cada una de las poses.

1. Doble bíceps de frente

De frente a los jueces, con una pierna de 40 a 50 centímetros hacia delante y a un lado, el atleta levanta sus brazos a nivel de los hombros doblando los codos, las manos deben estar cerradas con el fin de provocar la contracción de los bíceps y antebrazos, que son los principales grupos evaluados en esta pose. Adicionalmente, el competidor debe contraer tantos músculos como sea posible, ya que los jueces evalúan toda su estructura de la cabeza a los pies.

2. Expansión dorsal de frente

De frente a los jueces, con los pies y piernas en línea, manteniendo una separación de los talones de hasta 15 centímetros, el competidor colocará las manos en la parte baja de la cintura con los puños apretados, contra o agarrando la misma; en esta posición procurará ampliar al máximo los músculos dorsales. Al mismo tiempo, debe contraer tantos músculos como sea posible, para que su evaluación sea completa. Queda

Caranqui OE6-110 y Av. Mariscal Sucre Edificio FEGAL 3er. Piso La Magdalena Quito Ecuador
E-mail: csff.ifbb@gmail.com Teléfonos: 00593 2 6020107 00593 999614465
www.csff-ifbb.com

Confederación Sudamericana de Fisico Culturismo y Fitness

Reconocida por la IFBB Internacional - 2006

Argentina • Aruba • Bolivia • Brasil • Chile • Colombia • Ecuador • Paraguay • Peru • Uruguay • Venezuela

terminantemente prohibido que el atleta se agarre de la malla de posar y tire de ella hacia arriba o adelante, con el fin de mostrar la parte superior interna de los cuádriceps.

3. De perfil, expansión pectoral

El competidor puede elegir cualquiera de los lados en esta pose, con el fin de mostrar su mejor perfil. Se ubicará de lado doblando el brazo más cercano a los jueces en ángulo recto, con el puño cerrado, y con la otra mano sujetando la muñeca. La pierna que está hacia los jueces se mantendrá con la rodilla flexionada hacia adelante y apoyada en los dedos de los pies. A continuación el atleta, expandirá el pecho con la presión ascendente del brazo doblado al frente y contraerá el bíceps tanto como sea posible. Debe también contraer los músculos de las piernas, en especial el bíceps femoral; y por la presión en los dedos de los pies, se mostrará los músculos de la pantorrilla.

Confederación Sudamericana de Fisico Culturismo y Fitness

Reconocida por la IFBB Internacional - 2006

Argentina • Aruba • Bolivia • Brasil • Chile • Colombia • Ecuador • Paraguay • Perú • Uruguay • Venezuela

4. Doble bíceps de espalda

De espaldas a los jueces, el competidor deberá doblar los brazos y ajustar las manos como se detalla en la pose doble bíceps de frente, colocando un pie hacia atrás, apoyado fuertemente en los dedos, contraerá los músculos de las dos piernas, en especial los gemelos, bíceps femoral y glúteos. Debe mantener la contracción de los músculos de los brazos, acoplados con los músculos de los hombros, espalda superior e inferior, buscando una expansión de la estructura muscular superior.

5. Expansión dorsal de espalda

Ubicado de espaldas a los jueces, el competidor colocara las manos en la cintura con sus codos proyectados hacia afuera, con los pies y piernas en línea, manteniendo una separación de los talones de hasta 15 centímetros; entonces expandirá el dorsal lo más ancho posible. El atleta debe hacer un esfuerzo para mostrar contraídos tantos músculos como le sea posible. Queda terminantemente prohibido sujetarse de la malla de posar y tirar de ella hacia arriba o adelante con el fin de mostrar los músculos de los glúteos; así como descubrirse a propósito el glúteo.

Confederación Sudamericana de Fisico Culturismo y Fitness

Reconocida por la IFBB Internacional - 2006

Argentina • Aruba • Bolivia • Brasil • Chile • Colombia • Ecuador • Paraguay • Peru • Uruguay • Venezuela

6. De perfil. extensión de tríceps

El competidor puede elegir cualquiera de los lados para realizar esta pose, con el fin de mostrar su mejor perfil. Se ubica de costado hacia los jueces y coloca los brazos detrás de la espalda, debe obligatoriamente realizar un agarre entre los dedos o sujetar la muñeca o el brazo de adelante con la mano posterior. La pierna más alejada de los jueces estará flexionada hacia atrás con el pie apoyado en los dedos. El atleta ejercerá presión en su brazo de adelante, haciendo que se contraiga el tríceps; también contraerá el pecho, abdominales, muslo, glúteo y pantorrilla.

7. Abdominales y muslos

De frente a los jueces, el competidor coloca ambas manos detrás de la cabeza y una pierna adelantada, o con los pies y piernas en línea manteniendo una separación de los talones de hasta 15 centímetros (opcional). Llevando el tronco ligeramente hacia adelante el atleta contraerá los músculos abdominales y al mismo tiempo los músculos de las piernas.

Caranqui OE6-110 y Av. Mariscal Sucre Edificio FEGAL 3er. Piso La Magdalena Quito Ecuador
E-mail: csff.ifbb@gmail.com Teléfonos: 00593 2 6020107 00593 999614465
www.csff-ifbb.com