Francisco Revson Fernandes Pereira

Relatório de Estágio

Campina Grande, Brasil

7 de agosto de 2014

Francisco Revson Fernandes Pereira

Relatório de Estágio

Relatório de Estágio submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para obtenção do título de Graduado em Engenharia Elétrica.

Univesidade Federal de Campina Grande - UFCG

Centro de Engenharia Elétrica e Informática - CEEI

Departamento de Enegenharia Elétrica - DEE

Orientador: Edmar Candeia Gurjão

Campina Grande, Brasil

7 de agosto de 2014

Francisco Revson Fernandes Pereira
Relatório de Estágio/ Francisco Revson Fernandes Pereira. – Campina Grande, Brasil, 7 de agosto de 2014-
18 p. : il. ; 30 cm.

Orientador: Edmar Candeia Gurjão

Relatório de Estágio – Univesidade Federal de Campina Grande - UFCG
Centro de Engenharia Elétrica e Informática - CEEI
Departamento de Enegenharia Elétrica - DEE , 7 de agosto de 2014.

Francisco Revson Fernandes Pereira

Relatório de Estágio

Relatório de Estágio submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para obtenção do título de Graduado em Engenharia Elétrica.

Trabalho aprovado. Campina Grande, Brasil, 7 de agosto de 2014:

Edmar Candeia Gurjão
Orientador

Professor
Convidado

Campina Grande, Brasil
7 de agosto de 2014

Este trabalho é dedicado à minha família.

Agradecimentos

Os agradecimentos deste trabalho são direcionados:

- A Deus, pela força e perseverança, pela capacidade e sabedoria, por este mundo incrível... por tudo;
- Ao professor Edmar Candeia pela orientação no estágio e por me mostrar que respeito é umas das coisas mais importantes, tanto no meio acadêmico quanto na vida;
- A Rubem Medeiros pela ajuda na realização do estágio e por sempre estar disponível em tirar minhas dúvidas, até nas madrugadas;
- À minha família, pelo apoio, estímulo e por nunca desistirem de mim, não apenas durante esses cinco anos e meio, mas durante toda a minha vida;
- A todos os amigos e colegas conquistados nestes cinco anos de curso e durante a vida, por me mostrarem que, independente da situação e do momento, sempre é possível obter alegria e risos;
- A todos os professores e a todas as pessoas anônimas que, de forma direta ou indireta, contribuíram para a realização deste trabalho.

"A vida é uma peça de teatro que não permite ensaios. Por isso, cante, chore, dance, ria e viva intensamente, antes que a cortina se feche e a peça termine sem aplausos."

Charles Chaplin

Resumo

Neste trabalho é apresentado o desenvolvimento de uma interface para o sistema de processamento de áudio do Wireless Digital Mixer (WDM). Foi feita a criação de uma interface gráfica desenvolvida em Unity, essa interface gráfica consta da junção de parte do trabalho realizado, que foi a criação de interfaces gráficas para três tipos de efeitos de som, durante o estágio com alguns elementos já existentes e criados pela empresa, tais como os efeitos de ganho e pan. A segunda parte da interface desenvolvida foi a comunicação sem fio com o módulo Presonus 1818VSL, utilizou-se o protocolo de comunicação Open Sound Control (OSC) para que as interações feitas na pelo usuário do sistema na interface gráfica fossem transmitidos para esse módulo. Todos os dados usados durante a utilização do sistema, tais como os parâmetros dos filtros disponíveis, também podem ser guardadas e lidos por arquivos XML (eXtensible Markup Language) por meio de métodos de classes criados.

Palavras-chaves: Wireless Digital Mixer, Interface Gráfica, Protocolo de Comunicação Open Sound Control, Arquivos XML.

Abstract

This work presents the development of an interface to a processing audio system Digital Wireless Mixer (WDM). This was accomplished by creating a graphical interface developed in Unity, this graphical interface consists of the junction of the work done during the internship with some elements existing and created by the company. The second part of the interface has been developed for wireless communication with the Presonus 1818VSL module, we used the communication protocol Open Sound Control (OSC) for the interactions made by the user of the system in graphical user interface were transmitted for this module. All data used during system operation, such as the parameters of the filters available, can be saved and are also readable by XML files through methods created.

Key-words: Wireless Digital Mixer, Graphical Interface, Communication Protocol Open Sound Control, XML Files.

Lista de ilustrações

Lista de abreviaturas e siglas

CSOUND	C-Based Audio Programming Language
dB	Decibel
HPF	Filtro Passa-Altas
IP	Internet Protocol
OSC	Open Sound Control
PCM	Pulse Code Modulation
Q	Fator de Qualidade
RF	Rádio Frequência
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WDM	Wireless Digital Mixer
XML	eXtensible Markup Language

Sumário

1 Introdução

O objetivo do projeto da empresa no qual o estágio está relacionado é o desenvolvimento de um sistema de mixagem de áudio capaz de processar o sinal de áudio de maneira digital e de dispensar o uso de cabos em sua operação.

A grande quantidade de cabos e conectores que interligam os consoles de mixagem ao palco e aos equipamentos acessórios demandam o uso de estruturas de proteção adicionais (proteção mecânica, contra água, entre outros) para garantir a confiabilidade do sistema de sonorização. Essa necessidade provoca um aumento do tempo e custo de montagem e desmontagem do sistema de sonorização, e pode causar incômodo à plateia, uma vez que o multicabo fica no chão aos pés do público. A dimensão e peso desses equipamentos dificultam a logística de deslocamento de shows e aluguel de equipamentos, exigindo tempo e aumento de custos.

Há ainda um custo associado à manutenção desses sistemas mecânicos de conexão e cabos, uma vez que os mesmos sofrem danos severos devido ao transporte e à movimentação da plateia que inevitavelmente acaba pisando no multicabo. Um dano severo no multicabo no momento de um show inviabilizaria a sua continuidade. Outro problema no sistema convencional com fios é o custo do multicabo, o qual é proporcional à distância entre o palco e a ilha de mixagem. Em eventos de grande porte, o custo desse multicabo delimita o posicionamento da ilha de edição, podendo levá-la a um posicionamento inadequado.

Além dos custos, há ainda outras limitações apresentadas pelos sistemas convencionais, uma delas relaciona-se à expansibilidade do mesmo, ou seja, caso o mixer tenha X canais, e o evento demande $X + 1$ canais, o sistema não poderá ser expandido. Existem soluções de expansão no número de canais do mixer, contudo, no sistema convencional também seria necessário expandir o número de canais disponíveis no multicabo, o que só é possível por meio da aquisição de outro multicabo, implicando assim na necessidade de um investimento maior.

Pelos aspectos ditos, a LERJ Engenharia decidiu então desenvolver e comercializar a linha de produtos WDM (*Wireless Digital Mixer*), cujos principais aspectos tecnológicos inovadores do produto são baseados na aplicação de tendências tecnológicas existentes em seu estado da arte.

1.1 A Empresa

A LERJ Engenharia LTDA, que reside em Campina Grande-PB, é uma empresa que foi constituída pelos seguintes indivíduos: Erik Farias da Silva, Luiz Paulo de Assis Barbosa, Rubem José Vasconcelos de Medeiros e Olympio Cipriano da Silva Filho. A visão da empresa é se tornar uma referência na criação de soluções inovadoras em tecnologia da informação e comunicação para o mercado de áudio profissional. A empresa é um spinoff acadêmico, criada por profissionais com mestrado em Engenharia Elétrica pela Universidade Federal de Campina Grande (UFCG). A motivação para a criação da empresa decorreu da identificação de demandas provenientes do setor de áudio profissional, além da identificação, por parte dos sócios, de nichos de mercado carentes de soluções inovadoras. A empresa possui grande proximidade com instituições de ensino superior e técnico.

Seus sócios desenvolveram e desenvolvem projetos de alta relevância tanto no IECOM (Instituto de Estudos Avançados em Comunicações), que é lotado na UFCG e possui corpo composto por profissionais da instituição, quanto no IFPB (Instituto Federal de Educação, Ciência e Tecnologia da Paraíba) e na UFRN (Universidade Federal do Rio Grande do Norte). Essa proximidade entre a empresa, universidade e instituto de ensino possibilita que aquela esteja sempre em sintonia com as novas tecnologias e novos conhecimentos que surgem a cada dia. Isso se reflete diretamente nas áreas de atuação da empresa que estão em sintonia com os conhecimentos adquiridos na universidade, como é o caso do projeto proposto.

1.2 Objetivos do Estágio

O trabalho desenvolvido no qual este relatório está relacionado, teve como objetivo a ampliação de uma interface gráfica existente na empresa, fazendo com que a interface gráfica final ficasse mais próxima do tipo de produto que a empresa deseja. Também foi feito a comunicação entre a interface gráfica e o módulo referente a comunicação sem fio. Por fim, é mostrado que os parâmetros utilizados em tempo real pelo sistema podem ser salvos em arquivos XML e lidos posteriormente em classes desenvolvidas.

Este trabalho está dividido da seguinte forma: No Capítulo 2, é apresentado o sistema em desenvolvimento na empresa. No Capítulo 3, o treinamento e as atividades realizadas na empresa são mostradas.

2 Descrição do Sistema WDM

2.1 Interface Gráfica

A interface gráfica foi feita no Unity. Unity ou Unity 3D é uma ferramenta comercial multi-plataforma voltada para o desenvolvimento de jogos 3D. Esta ferramenta permite trabalhar com Direct X ou Open GL e suporta modelos 3D, animações, texturas, scripts e sons criados em outros programas para importação. Também suporta leitura de modelos 3D em vários formatos diferentes de arquivos. Possui suporte para jogos em rede com vários usuários, com scripts de sincronização e chamadas de procedimentos remotamente para os clientes disponíveis.

Unity suporta três linguagens de script: JavaScript, C# e um dialeto de Python chamado Boo. Normalmente a interpretação dos scripts é lenta, mas utilizados na ferramenta elas são compilados para código nativo e são quase tão rápidos quanto C++.

A ferramenta possui opção de diversas ferramentas já desenvolvidas, isto ajuda no desenvolvimento de interfaces gráficas para o sistema que será descrito posteriormente. Ela também possibilita que os projetos criados sejam construídos para diversas plataformas, tanto Windows e Linux quanto Android.

2.2 Especificação do Hardware WDM

O hardware do WDM será composto por uma parte analógica, uma parte digital, e um módulo RF. Seus subcomponentes são descritos abaixo.

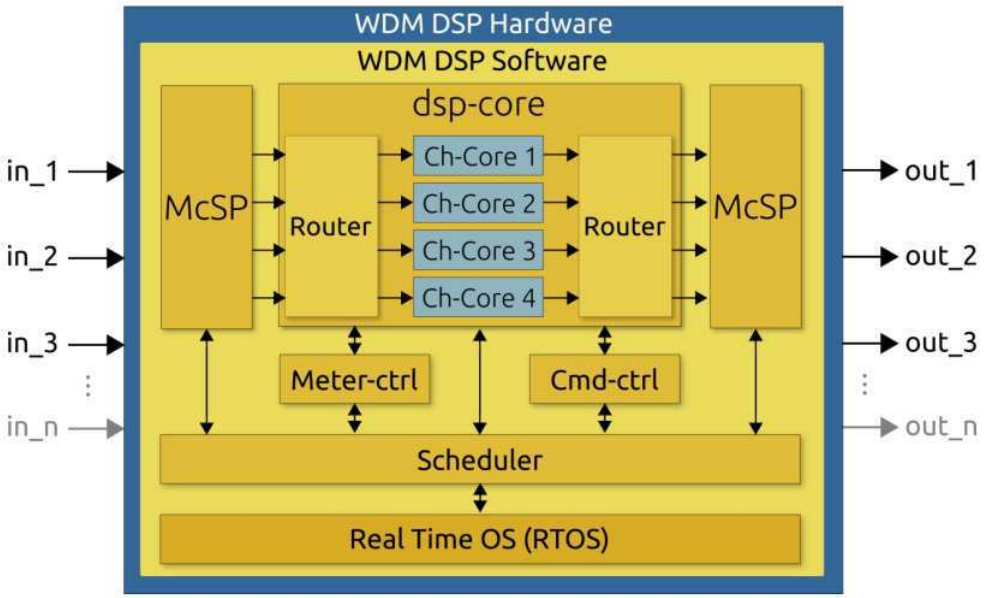

Figura 1 – Representação do Sistema WDM.

1. **Fase** - Inverte a fase do sinal (0-180°).
   - Parâmetros: *On*/*Off*.
2. **Atenuador/Ganho** - Diminui/Aumenta a intensidade do sinal.
   - Parâmetros
     - **Atenuador (*On*/*Off*)** - Quando ligado aplica uma atenuação em dB de valor constante.
     - **Ganho** - Grandeza em dB ajustável pelo usuário que deverá ser aplicada como ganho ao sinal de entrada.
3. **Filtro Passa Altas (HPF)** - Usado para eliminar frequências muito baixas presentes no sinal de entrada que são indesejáveis.
   - Parâmetros
     - **Frequência de corte** - Ajustável pelo usuário.
     - **Ganho** - Grandeza em dB ajustável pelo usuário que deverá ser aplicada como ganho ao sinal de entrada.
4. **Equalizador 4-bandas (Paramético)** - Usado para equilibrar determinadas faixas de frequências do sinal de entrada, usado após o HPF. Consiste em um banco de filtros podendo ser, quatro filtros passa faixas ou um conjunto de passa baixas, dois passa faixa e um passa altas.
   - Parâmetros

  - **Frequência de corte de cada filtro** - Ajustável pelo usuário.
  - **Banda passante de cada filtro (Q)** - Ajustável pelo usuário.
  - **Ganho de cada filtro** - Grandeza em *dB* ajustável pelo usuário que deverá ser aplicada como ganho ao sinal de entrada.

5. ***Gate*** - Bloqueia sinais de acordo com sua intensidade.

   - Parâmetros
     - ***Threshold*** - Intensidade em *dB*, ajustável pelo usuário. Sinais com intensidade menor ou igual ao *threshhold* serão bloqueados.
     - ***Ratio*** - Taxa de variação da atenuação aplicada aos sinias de intenside menor que *threshhold*.
     - ***Range*** - Faixa de intensidade para a qual a função de *gate* será aplicada. Este parâmetro ajusta a intensidade inicial da faixa e o *threshhold* ajusta a intensidade final da faixa [*range*, *treshold*]. Grandeza em *dB* ajustável pelo usuário.
     - ***Atack*** - Período no qual o gate está aplicando um ganho ao sinal de saidá para acompanhar o nível determinado pelo sinal de entrada. Aplica-se quando o sinal tem intensidade maior que *threshold*.
     - ***Release*** - Período no qual o gate está aplicando uma atenuação ao sinal de entrada para bloqueá-lo. Aplica-se qundo a intensidade do sinal de entrada foi menor que *threshold*.

6. **Compressor** - Usado para controle da faixa dinâmica do sinal por meio do aumento da intensidade de sinais fracos e pela diminuição da intensidade de sinais fortes.

   - Parâmetros
     - ***Threshhold*** - Intensidade em *dB*, ajustável pelo usuário. Sinais com intensidade maior ou igual ao *threshhold* serão atenuados.
     - ***Ratio*** - Taxa de variação da atenuação aplicada aos sinias de intenside maior que *threshhold*.
     - ***Ganho*** - Aplicado para equilibrar a intensidade do sinal após a atuação do compressor.
     - ***Knee*** - Diz respeito a transição que ocorre, na intensidade determinada por *threshold*, de sinais não comprimidos para sinais comprimidos. O parâmetro *knee* suavisa esta transição qundo ligado (*soft knee*) ou deixa a transição brusca quando desligado (*hard knee*).
     - ***Atack*** - Período no qual o compressor está aplicando uma atenuação ao sinal de entrada para atingir o nível determinado pelo parâmetro *ratio*. Aplica-se quando o sinal tem intensidade maior que que *threshold*.

  - ***Release*** - Período no qual o compressor está aplicando um ganho ao sinal de entrada para atingir o nível determinado por *ratio*, ou para 0 dB. Aplica-se quando a intensidade do sinal de entrada for menor que *threshold*.

7. ***Input Delay*** - Proporciona um atraso no sinal de entrada.

   - Parâmetros
     - **Atraso** - Tempo de atraso a ser aplicado no sinal em ms.

2.3 Sistema de Comunicação sem Fio

O sistema de comunicação sem fio é o WDMdsp. Ele é uma plataforma para processamento de áudio composto por software e hardware. Este documento contém detalhes técnicos da plataforma.

As características funcionais da plataforma são as seguintes:

- Múltiplos canais para processamento de efeitos de áudio
- Roteamento e Mixagem
- Controle do sistema através de mensagens OSC *(Open Sound Control)*

Figura 2 – Módulo Presonus 1818VSL.

2.3.1 Hardware

A entrada/saída do sistema feita por uma interface do módulo Presonus 1818VSL 24bit/96 kHz que é composta por

- 18x18 USB 2.0 *Audio interface*
- Conversores de 114 dB faixa dinâmica
- Taxas de amostragem de 48 KHz ou 96 KHz
- Resolução de 24 bits
- *Buffers*:
  - Mínimos: 64 @ 48 KHz , 128 @ 96 KHz
  - Máximos: 2048

Já o sistema de processamento é um ODROID-X2 com

- SoC: 1.7GHz Exynos4412 Prime Quad Core CPU
- Arquitetura: ARMv7
- Núcleos: 4
- Memória: 2G @ 400 MHz
- LAN: LAN9514 Hi-Speed USB hub
- Alimentação: 5V / 2A

2.3.2 Software

O software do WDMdsp é dividido em três componentes:

- *Processing Channel*
  - Canais independentes de processamento de áudio
    - Cadeia de instrumentos/efeitos CSOUND(1)
    - Efeitos podem ser ligados, desligados e inseridos dinamicamente
  - Jack
    - Roteamento de canais em tempo de execução
    - Mixagem de canais
- OSC Control
  - Mensagens de entrada para controle do sistema
  - Mensagens de saída para os monitores externos

- Linux-RT
  - Alsa: Driver de áudio PCM(2)
  - *Networking*: Driver de comunicação TCP/IP(3)
  - Controle de interrupções e escalonamento em tempo real (*hard*)(4)

2.3.3 Comunicação

Toda a comunicação com o WDMdsp é realizada por meio de mensagens OSC(5). Entidades conectadas à mesma rede que a plataforma podem enviar mensagens de controle assim como receber mensagens dos medidores.

2.3.3.1 Tipos de comandos OSC

- Controle de efeitos: /wdmdsp/effects
- Controle de canais: /wdmdsp/channels
- Roteamento: /wdmdsp/ports

2.3.3.2 Nomenclatura dos canais de processamento de áudio

Todos os canais de processamento de áudio possuem uma única entrada, e duas saídas: L(*left*) e R(*right*).

Os canais são referenciados nos seguintes formatos: chXXX:in, chXXX.outL, chXXX.outR, em que XXX é o número do canal contendo exatamente três casas decimais. Exemplos: ch001:in, ch001.outL, ch001.outR, ch019:in, ch019.outL, ch019.outR. A numeração dos canais começa sempre de 1.

2.3.3.3 Nomenclatura das entradas e saídas físicas

As entradas físicas são sempre denominadas por capture e as saídas por playback. Assim, um conjunto de entradas seria system:capture_1, $\cdots$, system:capture_18, e um conjunto de saídas seria system:playback_1, $\cdots$, system:playback_18.

2.3.3.4 Mensagens OSC

As mensagens OSC são enviadas no seguinte formato: oscsend <IP> <porta> <caminho> <formato> <mensagem>. Onde

- IP: É o endereço IP ou nome de rede da plataforma
  - O nome padrão em redes linux é visto como wdmdsp.local

- Porta
  - Porta usada para envio de mensagens de controle ao WDMdsp: 7770
  - Interessados em receber mensagens de medidores, deverão escutar a porta 5550
- Caminhos
  - /wdmdsp/effects: Envio de comandos para canais de processamento de áudio/efeito
  - /wdmdsp/ports: Envio de comandos de conexão de portas
  - /wdmdsp/channels: Envio de comandos de ativação/desativação de canais
  - /wdmgui/meter/in: Recepção de valores de medidas das entradas, a porta utilizada é a 5550
  - /wdmgui/meter/out: Recepção de valores de medidas das saídas, a porta utilizada é a 5550
- Formato da mensagem
  - isf: inteiro, seguido de string, seguido de float
    - Usado no caminho /wdmdsp/effects
  - sss: string, string, string
    - Usado no caminho /wdmdsp/ports
    - Três tipos de conexão
      - c1: entrada física -> canal de áudio
      - c2: canal de áudio->canal de áudio
      - c3: canal de áudio->saída física
  - is: inteiro seguido de string
    - Usado no caminho /wdmdsp/channels

Um exemplo da mensagem eviada é: oscsend 192.168.0.105 7770 /wdmdsp/channels isis 1 on 2 on. Onde foi suposto que o IP do WDMdsp seja 192.168.0.105, esse comando envia uma mensagem OSC para o caminho /wdmdsp/channels requisitando a ativação dos canais de processamento 1 e 2.

2.3.4 Efeitos

Os efeitos que o sistema pode reproduzir são: inversão de fase, ganho/atenuação, atraso do sinal, filtro passa-altas, *noise gate*, compressor, equalizador de 4 filtros, atraso com realimentação ao sinal, *reverb*. Além dos parâmetros de controle dos efeitos, todos os canais possuem dois parâmetros independentes de efeitos:

- "pan" - recebe valores de 0 a 1 e define o posicionamento estéreo do canal. 0 coloca o canal 100% na esquerda e 1 coloca o canal 100% na direita.
- "mute" - recebe os valores 0 ou 1. Quando 1, cancela a saída de som do canal e, quando 0, a libera normalmente

Todos os efeitos possuem quatro parâmetros denominados "xOn", "xN", "xIn" e "xOut", onde x é o nome do efeito.

- "xOn" - indica se o efeito está ativo ou inativo (em *bypass*) e recebe os valores 1 (ligado) e 0 (*bypass*).
- "xN" - indica o número de identificação (ID) atribuído ao efeito. Enviar um novo ID para um efeito faz com que ele seja automaticamente desligado e religado, o que pode causar "cliques" no som. Devido a restrições do CSOUND, um efeito deve ter sempre um ID menor que o dos efeitos que o seguem. Além disso, cada efeito só pode assumir IDs predeterminados. Os IDs que podem ser assumidos por um efeito são dados por 100*n + ID de base, onde n é um número natural. Assim, um efeito cujo ID de base é 3 pode assumir os valores de ID 3, 103, 203, 403, 503, 603, etc.
- "xIn" e "xOut" - indicam os canais de roteamento (internos ao CSOUND) dos efeitos. Estes canais são numerados de 0 a 10. O canal 0 é um canal "lixeira" onde os efeitos que não serão utilizados devem apontar suas entradas e saídas. O canal de entrada é o canal 1 e o de saída é o canal 10.

A interface gráfica é responsável por realizar a conversão entre o valor em decibel fornecido pelo usuário e o valor adequado para a mensagem OSC a ser enviada ao CSOUND.

3 Atividades Desenvolvidas

Serão apresentadas as atividades feitas durante o estágio no qual este relatório está relacionado. Para isso, será apresentada a fase de treinamento, onde ajudou ao estagiário a se familiarizar com a ferramenta Unity e com a linguagem de programação C#, conhecimento que foi utilizado em todo o decorrer do estágio. Posteriormente, os resultados do estágio serão mostrados, tanto a interface gráfica desenvolvida quando a implementação da comunicação sem fio com o módulo Presonus 1818VSL.

3.1 Treinamento Feito na Ferramenta Unity

Antes de começar as atividades, foi feita uma leitura sobre o material contido no site onde reside a ferramenta Unity(6). A ênfase dada foi principalmente na parte de interação entre componentes que a ferramenta proporciona, isto possibilita que seja possível ações em certos componentes influenciar outros, por meio de eventos. Eventos são formas de uma classe prover uma notificação ao cliente daquela classe quando alguma ação interessante acontece a um objeto, onde o cliente da classe pode ser outra classe. Também foi importante aprender a criar os objetos pré-fabricados, denominados *prefabs*. Eles permitem que o usuário do Unity guarde um *GameObject* completo com componentes e propriedades. Os prefabs podem ser usados como botões e barras de rolagem na interface gráfica.

Figura 3 – Tela padrão encontrada no Unity.

A interface desenvolvida no treinamento é apresentada na Fig. 4. Ela consta de dois botões centrais nos quais são limitados a andarem dentro do retângulo branco grande. Isto foi possível por meio de elementos delimitadores invisíveis criados no programa. Os botões também mostram sua posição em relação as coordenadas do retângulo branco grande.

Esta atividade também consta de mais dois botões. O botão à esquerda simula um botão de volume de áudio, onde a informação da sua posição é também mostrada ao seu lado direito. Por fim, o botão mais acima tem a característica de mudar de coloração toda vez que é clicado e manda um print para o prompt do Unity com uma informação lógica que é mudada toda vez que é clicado. Este último botão foi criado com o objetivo de se aprender como é feita a mudança de cor de um objeto sujeita a alguma ação externa que depende do usuário.

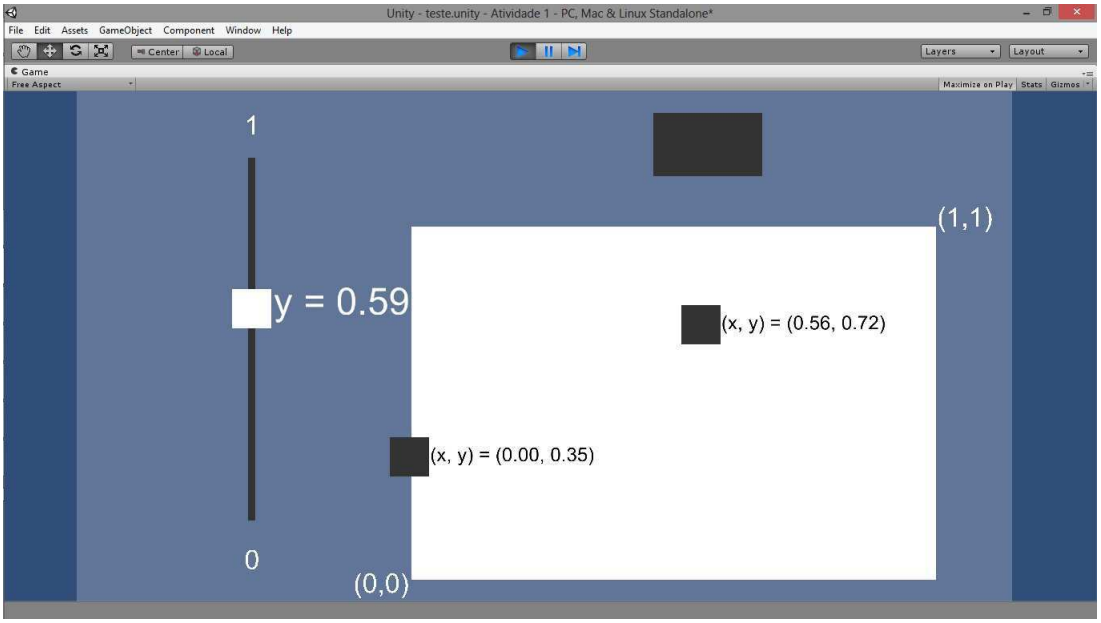

Figura 4 – Interface gráfica desenvolvida durante a fase de treinamento.

3.2 Interface Gráfica Desenvolvida

A primeira interface criada com o propósito de integrar o sistema final representa o equalizador de 4 filtros. Ela é apresentada na Fig. 5. Os quatro filtros podem ser passa-baixas, passa-faixa ou passa-altas. Além disso, os filtros também podem alterar o seu fator de qualidade, fazendo com que tenham uma maior variação na frequência. Eles possuem valores limites impostos na interface e no sistema, variam de $20Hz$ a $20kHz$.

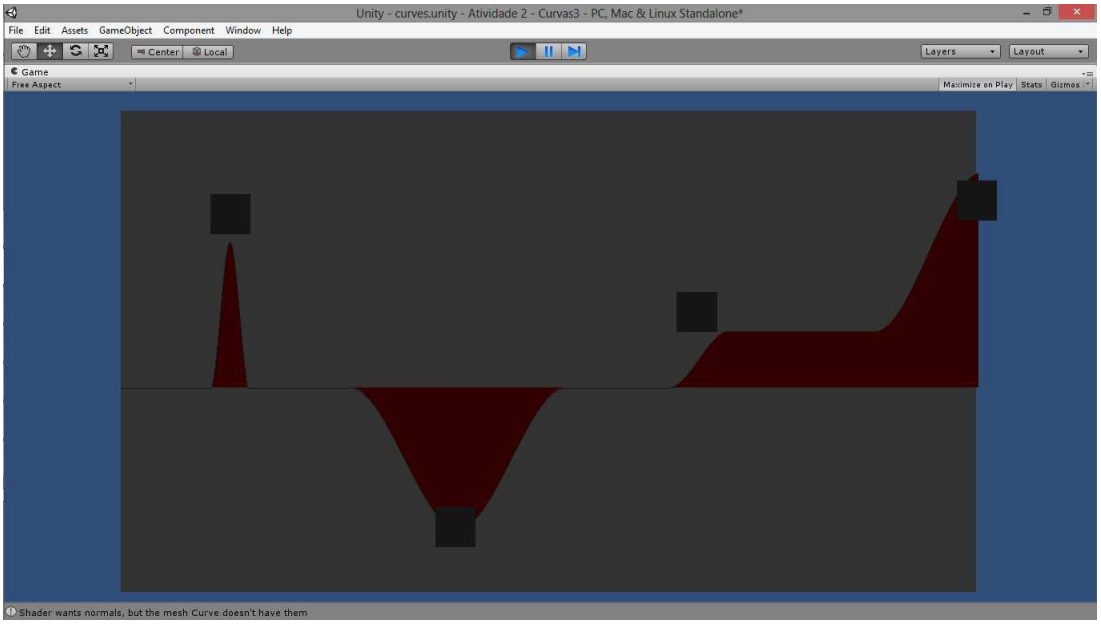

Figura 5 – Interface do Equalizador de 4 filtros desenvolvido.

Posteriormente, foi criado uma interface referente a um *noise gate* foi montada. Ela possibilita ao usuário altere tanto o valor do *threshold* do *noise gate* quanto do *range*, possibilitando que se elimine sons abaixo do volume desejado. A interface produzida é mostrada na Fig. 6.

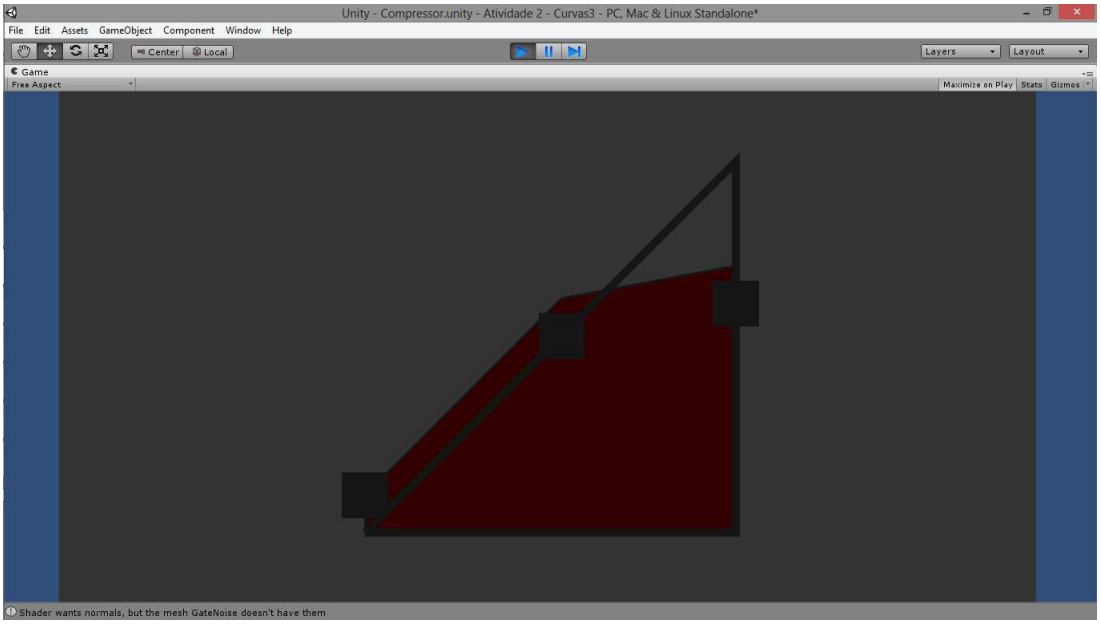

Figura 6 – Interface do Noise Gate desenvolvido.

O compressor também foi construído no Unity. A Fig. 7 expõe a interface desenvolvida. A liberdade da interface feita na implementação possibilita que sons com volume

superior a um determinado limite seja reduzido, enquanto os sons com volume mais baixo sejam amplificados.

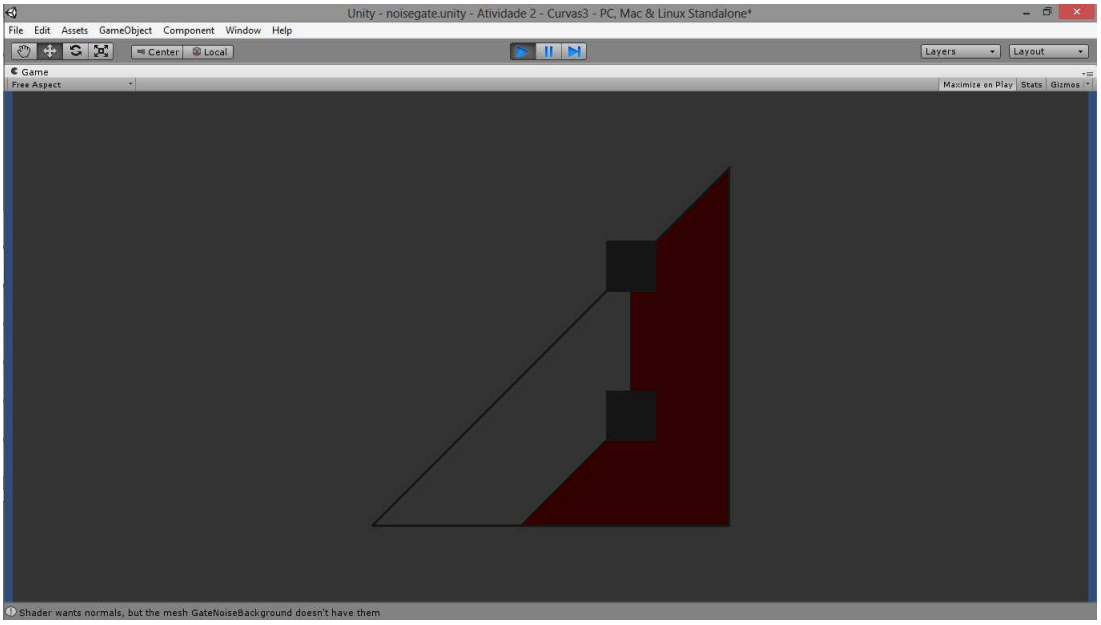

Figura 7 – Interface do Compressor desenvolvido.

Por fim, esses três efeitos foram integrados com os efeitos já construídos na empresa, que são ganho, pan e inversão de fase. Foi construído três botões na interface que dão acesso aos efeitos desenvolvidos no estágio. Cada canal que é disponível no sistema apresenta os três efeitos apresentados e eles são independentes entre si, ou seja, diferentes canais podem ter diferentes valores relacionados com qualquer um dos efeitos, os três já existentes mais os três criados no estágio referente a este relatório. A Fig. 8 mostra uma tela que tradicionalmente é vista pelo usuário quando estiver utilizando o sistema.

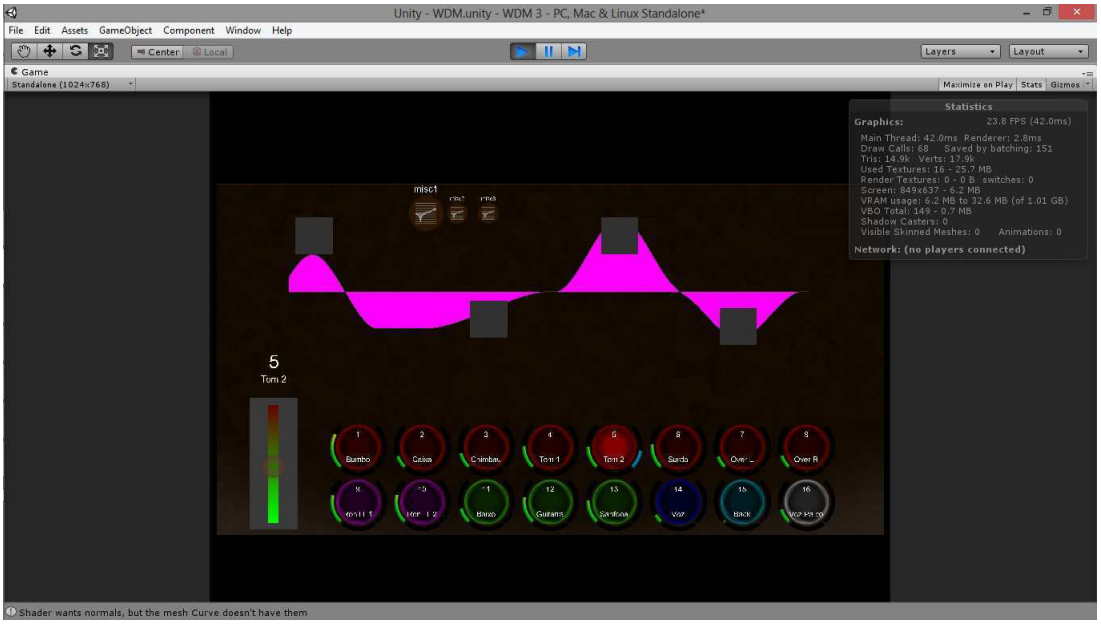

Figura 8 – Interface desenvolvida durante o estágio integrada com a já existente.

3.3 Implementação da Comunicação Sem Fio

Criado a interface para os efeitos, foi feito a parte da comunicação sem fio com o módulo Presonus 1818VSL. Ela foi feita com a utilização do OSC(5). O OSC é um protocolo de comunicação entre computadores, sintetizadores de som e outros dispositivos de multimídia que é otimizado para as modernas redes atuais. O tipo do protocolo de comunicação efetuado pelo OSC é UDP, o que é ótimo para o sistema em questão, pois a distância entre o dispositivo onde está a interface gráfica e o módulo Presonus 1818VSL deve ser relativamente pequena, fazendo com que a taxa de perda de pacote seja baixa. Outro aspecto importante para o uso de um protocolo UDP é a alta taxa de transmissão que é conseguida e a diminuição da complexidade computacional.

Como o OSC já efetua grande parte da programação de baixo nível, foi preciso apenas construir uma classe que fosse atualizada constantemente cada vez que houvesse atuação na interface e que o canal selecionado fosse mandado corretamente para o módulo. Um problema que se teve foi com a identificação dos efeitos, pois o módulo só aceita sistemas de identificação crescente de um efeito para o outro e cada efeito possui um conjunto de números de identificação pré-definido.

Para o uso do módulo, primeiramente foi feito um método que ligasse os canais de interesse. A seguir, um outro método efetua a açao de conectar as entradas físicas com os canais de áudio, os canais de áudio entre si, caso seja necessário, e os canais de áudio com as saídas físicas. Por fim, um terceiro método instancia os efeitos com os seus respectivos valores.

Uma vez que o sistema está em funcionamento, e não seja alterado nenhuma estrutura instanciada inicialmente, a ação do usuário em alterar qualquer um dos efeitos representará, para o sistema de comunicação sem fio, o recebimento de uma mensagem UDP com os novos valores dos efeitos que se deseja, sem precisar religar os canais ou as entradas e saídas.

3.4 Outras Atividades Realizadas

Além dessas atividades mencionadas, também foi construída uma classe responsável para carregar e salvar os parâmetros dos canais em XML. Isso possibilita que o usuário do sistema tenha suas próprias configurações iniciais e que, uma vez *setado* todo o sistema, ele possa utilizá-lo quantas vezes for desejado. Foi escolhido a linguagem XML por causa da sua comodidade, tradição e das suas ferramentas já construídas que auxiliam o desenvolvedor a construir estruturas complexas.

As atividades desenvolvidas no estágio seriam de difícil utilização por outro programador caso não houvesse um manual ou relatório descritivo. Assim, uma documentação do código escrito em C# foi escrita com a utilização do Doxygen. Ela possibilitou que todas as classes criadas pelo estagiário fossem referenciadas, documentadas e explicadas para qualquer próximo estagiário ou programador da empresa que venha a seguir.

4 Considerações Finais

Neste trabalho foi apresentado a interface construída durante a realização do estágio que este relatório descreve. A construção dessa interface seria impossível se não fosse realizado inicialmente um estágio de treinamento, e isso foi a primeira parte apresentada. O treinamento consistiu no desenvolvimento de uma interface simples com quatro elementos de interação.

Posteriormente, foi mostrado a primeira parte da interface desenvolvida, a interface gráfica. Ela constituiu de três elementos de efeitos criados durante o estágio mais três já existentes e pertencentes a empresa. Esses elementos foram integrados em apenas uma interface que constituirá a interface gráfica final do sistema desenvolvido pela empresa.

Também foi exposto o segundo componente de interface, o sistema de comunicação sem fio. Ele constituiu de classes construídas no Unity com a utilização do protocolo OSC para se comunicar com o módulo Presonus 1818VSL. Por fim, um sistema para ler e guardar os efeitos em para utilização ou em utilização, foi construído por meio da linguagem XML, possibilitando qualquer usuário do sistema construído à reutilizar as configurações impostas anteriormente pelo mesmo para os efeitos do sistema.

Sugestões para Trabalhos Futuros

Como trabalho futuro, será feito uma integralização dos efeitos construídos na interface com os que o módulo Presonus 1818VSL disponibiliza, mas que ainda não foi implementada. Também será feito um estudo sobre o desempenho de todas as classes utilizadas para que seja possível a otimização da interface.

Referências

1 BOULANGER, R. (Ed.). *The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing, and Programming*. [S.l.]: MIT Press, 2000. Citado na página 7.

2 GALLAGER, R. G. *Principles of Digital Communication*. [S.l.]: Cambridge University Press, 2008. Citado na página 8.

3 KUROSE, J. F. *Computer Networking: A Top-Down Approach*. [S.l.]: Pearson; 6th edition, 2012. Citado na página 8.

4 MANO, M. M.; KIME, C. *Logic and Computer Design Fundamentals*. [S.l.]: Prentice Hall; 4 edition, 2007. Citado na página 8.

5 UNITYOSC v1.1 October 2012 - Open Sound Control classes and API interface for Unity 3d. [S.l.]. Citado 2 vezes nas páginas 8 e 15.

6 Disponível em: <http://unity3d.com/>. Citado na página 11.