



**Modalidade do trabalho:** Relatório técnico-científico  
**Evento:** 2011 SIC - XIX Seminário de Iniciação Científica

## DESENVOLVIMENTO DE UMA FERRAMENTA GRÁFICA PARA A VISUALIZAÇÃO DA COBERTURA DE SINAL (WIRELESS) EM AMBIENTES COMPLEXOS<sup>1</sup>

**Eduardo Barbaro Netz<sup>2</sup>, Gerson Battisti<sup>3</sup>, Sandro Sawicki<sup>4</sup>.**

<sup>1</sup> Trabalho resultante de atuação de bolsista em projeto de pesquisa da Unijui

<sup>2</sup> BOLSISTA PIBIC/CNPq. Estudante do Curso de Ciência da Computação - eduardo.netz@yahoo.com.br

<sup>3</sup> Professor do Departamento de Ciência Exatas e Engenharias - battisti@unijui.edu.br

<sup>4</sup> Professor do Departamento de Ciência Exatas e Engenharias - sawicki@unijui.edu.br

### Resumo:

O projeto em desenvolvimento consiste na criação de um software onde o usuário poderá recriar a planta baixa de um ambiente qualquer e, com isso, poderá realizar a distribuição de AP (Access Point) neste ambiente. Com a distribuição pronta, o software irá pegar as posições determinadas dos APs pelo usuário e passará pela meta-heurística simulated annealing. Esta heurística tem o trabalho de redistribuir de melhor forma os APs dentro do ambiente. Após o cálculo das novas posições, o software deste trabalho deve apresentar os pontos de acesso nas novas posições. Esta heurística está sendo implementada por outro bolsista. Este artigo explica o processo de desenvolvimento de uma ferramenta gráfica. A ferramenta proporcionará o usuário na criação de ambiente de forma 2D e possível 3D, assim como na distribuição de APs com visualização sobre cada AP. Com ela, usuário está isento de todo processo de perturbação pois sua tela de visualização mostrará somente os APs sendo trocados de posição.

**Palavras-chave:** Visualização Gráfica, Open GL, Java 3D

### Introdução:

Uma questão essencial para garantir o desempenho e qualidade de redes sem fio é a análise da localização dos pontos de acesso. Este projeto tem como objetivo o desenvolvimento de um software para visualização do alcance das ondas de rádio em uma rede de computadores sem fio. O usuário poderá montar a planta baixa de seu ambiente, inserindo informações sobre a espessura da parede, o tipo de material usado na parede. Na etapa seguinte é possível fazer a distribuição dos APs (pontos de acesso wireless) livremente dentro do ambiente, mostrando também a abrangência do sinal do AP. O posicionamento final dos APs é determinado por outro programa que lê o posicionamento inicial e utiliza a heurística Simulated Annealing. O resultado da heurística irá determinar qual a melhor distribuição dos pontos dentro do ambiente de forma que haja o menor desperdício de sinal. Assim, a interface gráfica servirá para que um usuário leigo possa utilizar o programa sem precisar possuir conhecimentos técnicos de computação.



**Modalidade do trabalho:** Relatório técnico-científico  
**Evento:** 2011 SIC - XIX Seminário de Iniciação Científica

#### Metodologia:

O projeto iniciou com uma pesquisa e discussão com o orientador entre duas das tecnologias gráficas mais utilizadas. Baseando-se de um layout pré-definido o OpenGL e o Java3D são as que possuem mais ferramentas para o desenvolvimento e ambas atendiam as requisições. Com isto, o OpenGL foi a tecnologia gráfica escolhida para o desenvolvimento do software gráfico juntamente com a linguagem de programação C. Esta escolha foi feita pelo fato do OpenGL ser mais econômico em termos de processamento, uma vez que realiza um processamento e geração de imagens mais rápidas com menor uso de processador.

O OpenGL surgiu em 1992 por um grupo constituído por 3Dlabs, Apple Computer, NVIDIA, SGI e SUN. O OpenGL não é uma linguagem de programação, mas uma biblioteca de rotinas gráficas e de modelagens que possui mais de 250 funções. Infelizmente, o OpenGL não possui funções para gerenciamento de arquivos, para isto utiliza-se funções específicas do sistema operacional.

No início da pesquisa foram revisadas as linguagens C e C++ para que assim poder utilizar a o OpenGL. Para a utilização e aprendizado do OpenGL, foi utilizado o livro OpenGL – Uma Abordagem Prática e Objetiva de Marcelo Cohen e Isabel H. Manssour. Nele está descrito desde a instalação das bibliotecas e linkagens com elas, até a utilização de textura para objetos em três dimensões. A ferramenta desenvolvedora utilizada foi o Visual Studio 2010.

A aprendizagem do OpenGL foi baseada nos exemplos do livro citado acima, como a criação de um quadrado simples na tela, utilizando um comando “gl\_Begin”. Em seu bloco, são especificados a localização em que cada ponto será desenhado no plano cartesiano, ou seja “glVertex3f (0.0f,0.0f,0.0f)” indica que irá desenhado um ponto na coordenada cartesiana correspondente ao  $x=0$ , ou  $y=0$  e ao  $z=0$ . Seu parâmetro indica que tipo de desenho que irá ser formado, como por exemplo, o parâmetro “GL\_LINES” irá fazer a ligação entre os pontos desenhados. Assim, aos poucos mais e mais funções foram inseridas dentro do desenho (funções para desenho de um texto, funções para a rotação e a translação do desenho, o uso das setas do teclado como função de zoom) e, finalmente uma função para a alteração de tamanho da janela para que assim, quando a tela fosse redimensionada, o desenho também fosse redimensionado junto. Uma função importante utilizada foi a de animação, ela atualizava o desenho a cada movimento de rotação ou quando a função do zoom era chamada.

O Java3D possui uma estruturação diferente do OpenGL. Para a inicialização de um desenho qualquer, o Java3D cria um universo onde, dentro dele que o desenho tomará forma. Com o universo criado, pode-se começar o desenho. Ele possui maior facilidade na criação de desenhos, pois desenhos 3D podem ser instanciados passando por parâmetro seu tamanho. A estrutura passa a ser de um pai tendo vários filhos (BranchGroup). Os desenhos, ou seja, os filhos, devem ser adicionados ao universo (o pai), para que assim o desenho possa ser centralizado ou mesmo remodelado dentro do universo. Desta forma o Java3D pode ter mais de um universo.

#### Resultados e Discussão:





**Modalidade do trabalho:** Relatório técnico-científico  
**Evento:** 2011 SIC - XIX Seminário de Iniciação Científica

Inicialmente, o software desenhava um quadrado simples permitindo a digitação da posição no plano cartesiano de onde ele iria desenhar um círculo, chamando a função para desenho do círculo. Após este teste ter funcionado, passamos para a tentativa de sorteio da posição ao invés de ser digitado, o software sorteava uma posição dentro do quadrado especificado. Toda vez em que era pressionada a tecla de zoom ou mesmo a de rotação, estas chamavam a função anima que basicamente refaz todo desenho na tela com as variáveis atualizadas. Nos primeiros testes, esta função possuía um problema, que era o fato de ele “piscar” o desenho sem parada. Uma vez que a tela atualizava sempre que possível, pois era inicializada junto com a janela, a função de animação passou a ser chamada somente dentro de funções em que efetivamente realizam alguma alteração na visualização do desenho como a rotação ou zoom.

O mesmo procedimento, utilizando para múltiplos círculos, passava a sortear a posição fora da área determinada, assim quanto mais círculos possuía, mais o desenho do círculo se distanciava da área. Este problema se tornou bastante complicado (percebidos nos testes), quando a chamada a função de desenhar o círculo, ele passava como parâmetro a posição sorteada, porém para que o desenho estar no local de onde foi descrito, a função de desenho do círculo utilizava a função do OpenGL de translação do círculo. Como dito a função funcionava com apenas um círculo, mas quando passava a utilizar mais de um círculo a função deslocava não somente o segundo círculo atual, mas o primeiro também, e assim por diante com mais círculos. Infelizmente este problema ainda não foi solucionado, pois basicamente teria que tratar o novo círculo sempre como outra função.

Para tentar tornar a ferramenta gráfica mais amigável, buscou-se inserir uma função que utilizando o clique do mouse permitiria definir um ponto naquela coordenada, assim como o objetivo de desenhar retas ou ambientes inteiros, através do mouse. A proposta de desenho baseado na posição do mouse funcionou, porém as coordenadas cartesianas as quais normalmente são utilizadas, funcionam de forma diferente dentro do OpenGL, pois na janela não existe coordenada negativa, ou seja, o ponto (0,0) não está centralizado na tela mas sim no canto superior esquerdo. Assim a coordenada em que o botão do mouse, quando clicado, capturou, teria que ser convertido a coordenada cartesiana. Infelizmente depois de várias pesquisas, não se obteve resultado.

Outro fato importante determinado é que apesar de OpenGL ser uma ferramenta bastante rápida, infelizmente suas funções e criações de uma interface para o usuário era bastante complicado. Com isto tivemos de tomar a decisão de passar do OpenGL para a tecnologia java3D. Essa mudança permitiria obter uma interface com usuário melhorada, pois o Java possui uma coleção de interfaces prontas para serem usadas, sem que seja preciso uma programação mais detalhada.

No último semestre foi definido a mudança da linguagem OpenGL para Java 3D. Um ambiente de desenvolvimento foi montado, uma série de testes foram executados porém não foi suficiente para o desenvolvimento da ferramenta nessa nova linguagem.

Conclusões:





**Modalidade do trabalho:** Relatório técnico-científico  
**Evento:** 2011 SIC - XIX Seminário de Iniciação Científica

O projeto se mostrou um desafio e apesar de não tê-lo conseguido terminar, o projeto é muito interessante. Apesar dos esforços, o software se mostrou muito complexo e apenas alguns resultados foram conseguidos. Ainda haveria todo cálculo de propagação de sinal wireless em que o software deveria calcular e mostrar para o usuário. Portanto, apesar de tudo, o projeto acabou por ensinar e tirar muitas dúvidas sobre as linguagens de programação envolvidas no projeto.

#### Agradecimentos:

Meus agradecimentos vão aos professores Gerson Battisti e Sandro Sawicki que me fizeram o convite deste projeto. Também gostaria de agradecer ao CNPq pela minha aceitação no projeto e aos meus colegas bolsistas Marcelo Vanelli e Vinícius Schuster.

#### Referências:

COHEN, Marcelo, MANSSOUR, H. Isabel. OpenGL – Uma Abordagem Prática e Objetiva. São Paulo, Novatec, 2006.

SOULIÉ, Juan. C++ Language Tutorial , 2008. Disponível na internet em: <http://www.cplusplus.com/doc/tutorial>.

SELMAN, Daniel. Java 3D Programming. 2002. Disponível na Internet em: <http://www.manning.com/selman/>

MANSSOUR, H. Isabel. Java3D API. Disponível na internet em: <http://www.inf.pucrs.br/~manssour/Java3D>.

Henry A. Sowizral, David R. Nadeau. Introduction to Programming with Java 3D. Disponível na internet em : <http://www.javasoft.com> .