

LABORATÓRIO VIRTUAL DE FÍSICA

Thiago Bedin Frustaci¹, Vladimir Cavalcanti Dobroff², Roberto Scalco³

Abstract — This work presents the development of a virtual physics laboratory that can build and simulate rigid body dynamics physics scenes. The professors can create, edit, save and simulate experiments using 3D models stored in VRML file format. Later, students can simulate those experiments and use virtual measuring tools like a ruler, a protractor, a stopwatch and an object trajectory tracking plane to study the experiment. The professor and the student computer applications can be used to complement the learning of the traditional Physics I classes at the Mauá School of Engineering. These applications were developed using the Delphi language, the Open Dynamics Engine (ODE) library to simulate the physics and the Open Graphics Library (OpenGL) to render the scenes.

Index Terms — Open Dynamics Engine, OpenGL, Virtual Physics Laboratory, VRML.

INTRODUÇÃO

A evolução tecnológica atual permite que aplicativos complexos de computador sejam executados em tempo real modificando algumas das suas características de acordo com a interação do usuário.

Os jogos com ambientes tridimensionais são um ótimo exemplo, algumas vezes unindo poderosos sistemas gráficos com *engines* físicas realísticas, tendo como único propósito o entretenimento.

Entretanto, poucos percebem que exatamente a mesma tecnologia que dá vida a um jogo pode ser utilizada com propósitos diferentes, por exemplo, educacionais.

É comum encontrar sítios na Internet com aplicativos desenvolvidos em Java (*applets*) voltados para o aprendizado de física [1]. Porém, estes aplicativos muitas vezes não são “divertidos” o bastante para atrair a atenção dos usuários.

Desta maneira, é essencial que o aplicativo seja interativo o suficiente para motivar o usuário, sem esquecer de que o principal enfoque é o ensino de Física. Esta interatividade deve permitir ao usuário propor modificações, mesmo que pequenas, nas experiências, além de realizá-las efetivamente, observando e medindo os resultados imediatamente.

TRABALHOS RELACIONADOS

Esta seção apresenta um conjunto de projetos e aplicações que serviram como base para o desenvolvimento do laboratório virtual de física.

OpenGL

A *Open Graphics Library* (OpenGL) é uma biblioteca que permite a manipulação de dados em um espaço tridimensional [2]. Os resultados gráficos são obtidos rapidamente, uma vez que atua diretamente com o *hardware* gráfico, além de ser extremamente portátil.

Sua interface é composta por mais de 250 instruções que podem ser utilizadas para desenhar figuras primitivas como pontos, linhas e polígonos, em duas e três dimensões. Além disso, suporta efeitos como iluminação, mapeamento de textura, entre outros.

Engines de Física

A *Open Dynamics Engine* (ODE) é uma biblioteca com código-fonte aberto de alto desempenho capaz de simular em tempo real a dinâmica de corpos rígidos [3]. Além disso, possui um sistema de detecção de colisão integrado conhecido como OPCODE (*Optimized Collision Detection*).

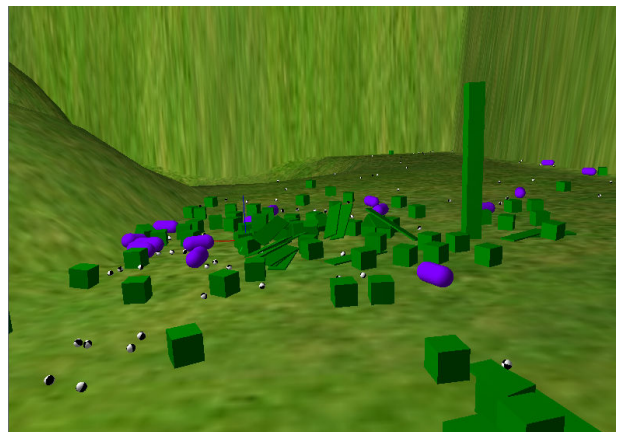


FIGURA. 1

SISTEMA DE COLISÃO COMPLETO COM CENAS DE COMPLEXIDADE CONSIDERÁVEL EM TEMPO REAL UTILIZANDO A ODE.

1 Thiago Bedin Frustaci, Centro Universitário do Instituto Mauá de Tecnologia – Escola de Engenharia Mauá, Praça Mauá, 1, 09.580-900, São Caetano do Sul, SP, Brasil, frustaci@gmail.com

2 Vladimir Cavalcanti Dobroff, Centro Universitário do Instituto Mauá de Tecnologia – Escola de Engenharia Mauá, Praça Mauá, 1, 09.580-900, São Caetano do Sul, SP, Brasil, vlad.dobroff@gmail.com

3 Roberto Scalco, Centro Universitário do Instituto Mauá de Tecnologia – Escola de Engenharia Mauá, Praça Mauá, 1, sala G-02, 09.580-900, São Caetano do Sul, SP, Brasil, roberto.scalco@maua.br

Como a biblioteca foi desenvolvida para ser utilizada em simulações em tempo real ou aplicativos que permitam a interação com o usuário, foi dada prioridade para algoritmos rápidos, porém relativamente estáveis em termos de cálculos. Para garantir um melhor desempenho visual, existem algumas simplificações nos algoritmos que podem gerar resultados com menor exatidão física.

A biblioteca suporta alguns primitivos nativamente, como: o paralelepípedo, a esfera e a cápsula. Além disso, possui uma estrutura conhecida como malha de triângulos (*triangle mesh*) que permite utilizar representações de quaisquer objetos que sejam constituídos exclusivamente por triângulos.

PhysX

No início de 2006 foi lançado pela empresa Ageia [4] a primeira unidade de processamento de Física ou PPU (*Physics Processing Unit*), que revolucionou essa área tecnológica fazendo com que outras empresas passassem a investir em soluções para física em tempo real.

A PPU é um processador dedicado criado especialmente para realizar o processamento de algoritmos de Física complexos em tempo real. A idéia da PPU é análoga ao da GPU (*Graphics Processing Unit*) especializada no processamento gráfico.

Para o desenvolvimento de aplicativos capazes de serem acelerados pela PPU é necessário utilizar a biblioteca PhysX, antiga NovodeX. Estes aplicativos funcionam normalmente caso o processador de física não esteja disponível no computador, entretanto, os resultados são obtidos mais lentamente, uma vez que todos os cálculos deverão ser realizados pela CPU.

As vantagens de se utilizar um processador dedicado são muitas, dentre elas, maior variedade de fenômenos físicos (dinâmica de corpos rígidos, deformáveis, fluídos), detecção de colisão, simulação de fios de cabelo, tecidos, análise de elementos finitos e fraturamento de objetos. Associado a todas estas possibilidades, menos sobrecarga da CPU e maior velocidade, complexidade e precisão da simulação física.

DESENVOLVIMENTO

O projeto principal é caracterizado pelo desenvolvimento de duas aplicações: do professor e do aluno. Ambas foram implementadas utilizando a linguagem Object Pascal sobre o ambiente de desenvolvimento Delphi 6.0. Além disso, foi utilizada a biblioteca gráfica OpenGL e a biblioteca de Física ODE.

Partes fundamentais para o funcionamento de ambos os aplicativos foram implementados separadamente como bibliotecas dinâmicas DLL (*Dynamic Linked Library*).

O desenvolvimento dos aplicativos em módulos DLL distintos facilitou a integração do projeto como um todo e permitiu a criação de três bibliotecas de programação:

- **Biblioteca de visualização da cena:** define uma câmera virtual para visualização tridimensional da cena. Implementada em Delphi 6.0.
- **Biblioteca de detecção de objetos:** determina quais são os objetos que o usuário está selecionando com o cursor do mouse. Implementada em Delphi 6.0.
- **Biblioteca leitora de VRML:** carrega os modelos 3D salvos em disco que serão exibidos graficamente. Implementada em C++.

Além disso, as cenas criadas no aplicativo do professor são armazenadas em disco no formato XML [5] para utilização posterior no aplicativo dos alunos.

Para a manipulação do arquivo de dados em XML foi utilizado o *XML Data Binding Wizard* presente na versão Enterprise do Delphi 6.0. Este *wizard* analisa arquivos XML quaisquer e gera o código-fonte orientado a objetos em Delphi para manipular tais arquivos, baseando-se na nomenclatura interna utilizada no arquivo XML analisado.

Para criar a cena, o professor pode utilizar os objetos básicos disponíveis como paralelepípedos, esferas, cápsulas e malhas de triângulos.

O novo objeto criado pelo professor pode ser simulado perfeitamente utilizando a malha de triângulos, mas também é possível que tal objeto seja representado internamente pela ODE como um de seus primitivos, fazendo uma aproximação do objeto quando for conveniente.

O professor pode adicionar novos objetos e alterar seus parâmetros físicos como posição, rotação, velocidade, massa, aceleração, força, entre outros.

A integração com a ODE é completa, de forma que também é possível interagir com a cena em tempo real quando a simulação física está em execução.

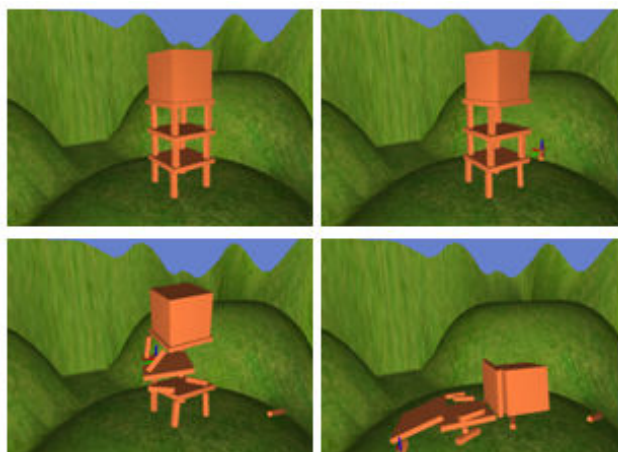


FIGURA. 2
CENA INTERATIVA EM TEMPO REAL.

As experiências elaboradas com o aplicativo do professor podem ser posteriormente abertas pelos alunos em

seu devido aplicativo. É permitido aos alunos visualizar e simular a cena criada, utilizando ferramentas virtuais de medição, como uma régua, um transferidor, um cronômetro e um plano que registra a trajetória dos objetos. Além disso, alguns parâmetros, definidos pelo professor durante a confecção da cena, podem ser alterados pelos alunos para que estes possam verificar a sensibilidade dos modelos físicos a determinadas grandezas.

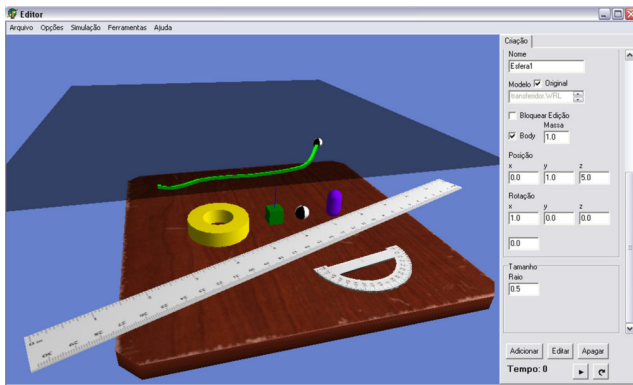


FIGURA. 3

FERRAMENTAS VIRTUAIS E OBJETOS GENÉRICOS DISPONÍVEIS.

A seguir uma descrição geral das bibliotecas desenvolvidas.

Câmera Virtual

Para que fosse possível implementar um laboratório virtual de Física, foi necessário criar uma forma prática para visualizar as cenas. Isso deveria ocorrer tanto durante o processo de modelagem quanto durante a simulação, quando a experiência estivesse sendo executada.

De forma a manter o projeto organizado, esta biblioteca foi desenvolvida separadamente, onde foram implementados procedimentos que simulavam a operação de uma câmera virtual em um ambiente tridimensional. Entende-se por câmera o posicionamento e alinhamento de um observador imerso em uma cena que está sendo visualizada.

A DLL disponibiliza sub-rotinas capazes de posicionar a câmera em um determinado local com uma rotação inicial, utilizando o sistema de coordenadas da cena. Além disso, é possível definir parâmetros referentes ao deslocamento e rotação relativos aos seus próprios eixos x , y e z .

Utilizando as funções implementadas em conjunto com a movimentação do mouse, foi possível facilitar a visualização do ambiente. Desta maneira, tornou-se possível rotacionar o campo de visão do observador da experiência pela simples utilização do mouse, sem que ocorressem restrições no seu posicionamento. Isso reduz a necessidade de edição de parâmetros durante a execução do aplicativo, tornando a sua utilização menos complexa.

A biblioteca também possui o recurso de múltiplos campos de visão ou múltiplas câmeras. Este recurso consiste

na definição destes campos e na alternância entre eles. Pode-se entender melhor este conceito imaginando um conjunto de câmeras posicionadas em um determinado ambiente. Cada uma gera uma imagem de acordo com o seu ponto de vista. Isto possibilita a observação da cena de diferentes ângulos sem a necessidade de deslocar nenhuma delas.

Apesar da implementação deste recurso, esta funcionalidade não é utilizada no projeto. Isso ocorre em virtude da necessidade de alterações não previstas inicialmente na estrutura do projeto, em especial no que se refere à leitura e gravação dos dados das experiências a partir dos arquivos no formato XML.

Deteccção de Objetos

Além da necessidade de visualizar a cena a partir de diferentes ângulos, também existe a necessidade de movimentar os objetos nela contidos de forma independente dos demais, assim como alterar os seus parâmetros. Por este motivo, tornou-se indispensável implementar uma biblioteca cuja finalidade seria auxiliar neste processo, permitindo a seleção dos objetos a partir do clique do mouse sobre os mesmos.

Em uma implementação inicial do projeto, apenas era possível adicionar elementos na cena, sem a opção de editá-los. Mesmo salvando a experiência em diferentes arquivos, de acordo com o seu estágio de montagem, não era possível realizar alterações significativas na montagem finalizada. Como um dos objetivos era exatamente facilitar a montagem e execução das experiências, era necessário modificar esta característica. Como uma solução inicial, além de possibilitar a exclusão de elementos e a edição dos seus parâmetros, a seleção dos objetos por meio de uma lista contendo as suas identificações foi cogitada. Apesar de ser uma solução válida, em cenas mais complexas, com uma quantidade considerável de objetos, haveria uma maior dificuldade para modificá-los. Considerando este fato e o grau de desenvolvimento do aplicativo, optou-se por desenvolver uma biblioteca capaz de auxiliar nesta tarefa, sem precisar realizar alterações significativas no restante do projeto. Por esta razão foi criada a DLL de deteção de objetos.

A DLL criada replica alguns dos comandos existentes na biblioteca gráfica OpenGL, de forma a registrar internamente os triângulos que estão sendo tonalizados na imagem. Para utilizá-la, a cada chamada de procedimento da biblioteca gráfica OpenGL é necessário utilizar o procedimento equivalente da biblioteca de deteção. Como a grafia e os parâmetros das funções replicadas são praticamente os mesmos (com exceção do parâmetro adicional que identifica o objeto), isso facilitou a sua integração com o restante do projeto. Além destas funções, existem outras que recebem os parâmetros da câmera ativa e as coordenadas do clique do mouse. Com a inserção destes parâmetros a biblioteca retorna a identificação do objeto selecionado.

O funcionamento da DLL se resume na criação de pilhas de vértices, matrizes (de rotação, deslocamento e mudança de escala) e comandos (réplicas da OpenGL), as quais são armazenadas para serem processadas apenas no momento que ocorrer o clique do mouse. No decorrer do desenvolvimento, também foi acrescentado o suporte às *display lists* da OpenGL, o que ampliou o uso da biblioteca. Para utilizá-la basta fornecer a mesma identificação para cada vértice (ou *display list*) pertencente a um mesmo objeto. Se o usuário do aplicativo clicar sobre qualquer triângulo pertencente a cena, seu número de identificação será retornado pela biblioteca.

Inicialmente a DLL apresentava um grande consumo de memória e tempo de processamento, o que levou a busca de alternativas quanto a sua utilização. Por esta razão verificou-se que o sistema de colisão existente na ODE também poderia ser utilizado para realizar a detecção dos objetos. Entretanto, este sistema possui algumas limitações, sendo uma delas referente à representação gráfica de determinadas geometrias. Se, por exemplo, uma esfera fosse representada internamente na ODE como um cubo (como uma aproximação), a detecção seria realizada sobre o cubo e não sobre a esfera, o que geraria problemas para objetos muito próximos. Este conceito poderia ser estendido no caso de geometrias compostas ou malhas de triângulos, caso a representação gráfica do objeto não fosse igual ao modelo utilizado pela ODE.

Posteriormente o problema do consumo excessivo de memória e tempo de processamento foi resolvido na DLL de detecção de objetos, permitindo a sua utilização. Como a DLL realiza a detecção do clique sobre os triângulos que estão sendo efetivamente tonalizados não ocorrem problemas para objetos muito próximos, pois a detecção é precisa. Outra vantagem da biblioteca seria o seu funcionamento independente da ODE ou da própria OpenGL.

Leitor de Arquivos VRML

Esta biblioteca foi desenvolvida para permitir importar qualquer modelo tridimensional salvo em disco no formato VRML.

A biblioteca realiza a leitura do arquivo, salvando na memória RAM os vértices, os seus vetores normais e as coordenadas de textura associadas, assim como os índices dos vértices, normais e coordenadas de textura na ordem em que formam o modelo 3D.

Após a leitura, é possível acessar os dados na memória, utilizando estruturas e funções apropriadas, para informar à biblioteca OpenGL as características dos modelos para serem corretamente tonalizados.

Devido ao padrão VRML ser bastante genérico [6], em programas como o MicroStation, por exemplo, o modelo exportado acaba contendo polígonos com número de vértices variados, isto é, não apenas triângulos.

Entretanto, para ser possível construir uma malha de triângulos na ODE é necessário que o modelo seja

representando apenas por triângulos. Associado a isto, a interface com a biblioteca gráfica OpenGL torna-se otimizada para modelos contém apenas triângulos.

Desta maneira, foi necessário não apenas ler o arquivo VRML, mas também realizar seu devido processamento com o objetivo de dividir polígonos com mais de 3 vértices em triângulos correspondentes.

Além disso, por razões de estabilidade no cálculo de Física e maior velocidade de processamento a biblioteca ODE recomenda que as malhas de triângulos sejam compostas por uma lista de vértices únicos e uma lista de índices na seqüência em que os vértices são ligados, formando os triângulos que compõe a geometria.

Por esse motivo, a biblioteca VRML tem a capacidade de eliminar os vértices com coordenadas iguais (x, y e z), reindexando a lista dos vértices. Consequentemente torna-se necessário reindexar triângulos separados para que formem uma malha de triângulos única.

Em virtude da liberdade do padrão VRML, o processamento do arquivo considera algumas particularidades geradas por diferentes aplicativos comerciais de modelagem, como o MicroStation v8.1, 3D Studio Max 6.0, MilkShape 3D 1.7 ou Blender 2.41. Embora o VRML não seja o formato padrão de nenhum destes aplicativos, cabe ao usuário exportar os modelos criados para o formato VRML.

RESULTADOS

Ao longo do desenvolvimento, foi dada prioridade para o desempenho das bibliotecas e dos aplicativos como um todo, já que estes podem vir a ser executados em computadores com configurações diferentes.

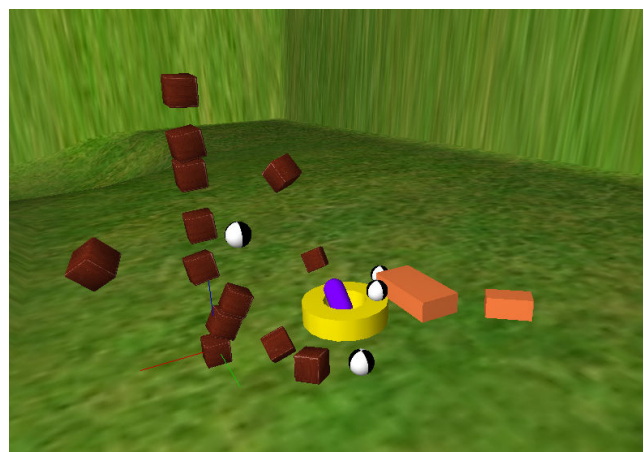


FIGURA. 4

DIVERSOS OBJETOS DIFERENTES INTERAGINDO.

A configuração mínima em que o programa foi testado é um processador AMD Athlon 1,0 GHz com 256 RAM DDR e uma placa de vídeo NVidia GeForce 4 MX.

Para verificar o comportamento dos modelos Físicos, uma experiência realizada pelos alunos no laboratório foi recriada e os resultados medidos pelas ferramentas do aplicativo foram comparados com diversas medidas realizadas *in loco* pelos alunos. As medições realizadas no laboratório para a experiência “colisão bidimensional” resultaram em valores entre 89° e 112° para o ângulo entre as trajetórias dos corpos. Ao utilizar as ferramentas desenvolvidas, foi obtido um ângulo de 108°.

CONCLUSÃO

Este projeto criou um laboratório virtual tridimensional de Física capaz de ser utilizado pelo professor para criar experiências personalizadas, interativas e em tempo real.

Com o uso do aplicativo, não se pretende eliminar a prática existente no laboratório, mas complementar o aprendizado dos alunos sugerindo, após a realização presencial da experiência, que esta seja repetida com algumas variações de parâmetros, como massa, dimensão ou velocidade inicial dos corpos.

Isto permite que os alunos possam desenvolver uma habilidade relacionamento entre os conceitos e modelos matemáticos da Física com os seus correspondentes no “mundo real”.

Ainda que limitado pela tecnologia atual disponível na biblioteca ODE, certamente o laboratório poderá ser adaptado para abranger um maior número de fenômenos físicos, com maior precisão física e desempenho de execução, assim que tecnologias como a PhysX se tornarem mais utilizadas e portáteis para outras linguagens de programação.

AGRADECIMENTO

Os integrantes do projeto agradecem ao Centro Universitário do Instituto Mauá de Tecnologia permitir que mais esta ação didática pudesse ser desenvolvida e aplicada.

REFERÊNCIAS

- [1] HWANG, F. K. NTNUJAVA: Virtual Physics Laboratory. Disponível em: <<http://www.phy.ntnu.edu.tw/ntnujava/>>. Acesso em: 16 ago. 2006.
- [2] OpenGL – The Industry Standard for High Performance Graphics, Disponível em <<http://www.opengl.org/>>. Acesso em: 01 set. 2006.
- [3] SMITH, R. ODE – Open Dynamics Engine. Disponível em: <<http://www.ode.org/>>. Acesso em: 26 fev. 2006
- [4] AGEIA. PhysX. Disponível em <<http://www.ageia.com/>>. Acesso em: 19 abr. 2006.
- [5] XML – Extensible Markup Language. Disponível em: <<http://www.w3.org/XML/>>. Acesso em: 17 jul. 2006.
- [6] WEB 3D CONSORTIUM. VRML Archives. Disponível em: <<http://www.web3d.org/x3d/vrml/>>. Acesso em: 07 abr. 2006.