

Trabalho de Conclusão de Curso Animação de Fenômenos Físicos

Rafael Issao Miyagawa Alberto Hideki Ueda
João Pedro Kerr Catunda

Orientador: José Coelho de Pina Junior

Novembro de 2012

Conteúdo

I	Parte objetiva	3
1	Introdução	3
1.1	Física no BCC	3
1.2	Problemas ao simular interações físicas	4
1.3	Simulador	4
1.4	Estrutura da Monografia	4
2	Plataforma computacional	6
2.1	Esquema de dependências	6
2.2	Ruby	6
2.3	Simulação com Chipmunk	6
2.4	Animação com Gosu	6
2.5	Cenários com Glade	7
2.6	Exemplo	7
3	Discretização da simulação	8
3.1	Tempo de simulação	8
3.2	<i>Time step problem</i>	8
3.3	Problema de granularidade	8
4	Colisões	9
4.1	Algoritmos para detecção	9
4.2	Colisões no Chipmunk	9
4.3	Efeito de colisões	9
5	Atividades realizadas	10
6	Animações produzidas	11
7	Physimulation	12
8	Introdução à computação com animações	13
9	Comentários finais	14
9.1	Trabalhos futuros	14
9.2	Desafios	14
9.3	Conclusão	14

10 Roteiro de instalação da plataforma	15
11 Apêndice	16
II Parte subjetiva	17
12 Alberto Ueda	17
12.1 Desafios e frustrações encontrados	17
12.2 Disciplinas mais relevantes	18
12.3 Estudos futuros	20
13 Rafael Miyagawa	20

Parte I

Parte objetiva

1 Introdução

O foco deste projeto é atrair a atenção dos alunos do IME em relação às disciplinas de física ministradas no curso de bacharelado em Ciência da Computação.

1.1 Física no BCC

Embora seja comum encontrar alunos da computação que tenham interesse em tópicos relacionados a física clássica e moderna - Leis de Newton, Leis de Kepler, Teoria da Relatividade de Einstein, entre outros - há considerável resistência do corpo discente em relação a grade curricular do BCC incluir as disciplinas de física. (TODO citar questionário BCC) Isto não se aplica apenas às disciplinas de física, mas também a álgebra, cálculo, probabilidade e estatística (TODO referência).

Visando compreender melhor a situação e apresentar uma proposta de reformulação da grade ou das disciplinas, foi criado o Grupo de Apoio ao BCC (TODO nomes), uma iniciativa da Comissão de Coordenação do BCC (CoC). Após a aplicação de questionários e conversas nos Encontros BCC (TODO citar), um dos consensos de ambas as partes - professores e alunos - foi o de que seria interessante uma maior contextualização destas disciplinas para os alunos em termos de computação. No apêndice da seção 1.1 há uma cópia de um texto produzido pelo CoC, com referências para o documento original.

Tal contextualização pode ser feita de várias formas: por parte dos professores das disciplinas - mostrando aplicações dos tópicos estudados para computação, trazendo professores ou especialistas em sala de aula que possam transmitir experiências relacionadas ao conteúdo aprendido - ou por parte dos próprios alunos - reunindo-se para palestras extra-classe, desenvolvendo bibliotecas que motivem os demais alunos a utilizar, pesquisar e até mesmo programar baseando-se nos conceitos adquiridos.

Nesse sentido, o Physimulation é uma ferramenta para simulação e animação de fenômenos físicos e tem como um dos principais objetivos integrar o conhecimento adquirido em FAP-126 (Física I) com um ambiente de programação (TODO programa?).

Com este trabalho visamos contextualizar os assuntos abordados na disciplina com demonstrações de ambientes físicos, integrações com exercícios-programas (EP's) e a apresentação de desafios atuais de simulação no campo da física.

1.2 Problemas ao simular interações físicas

Existem muitos problemas ao tentar simular um ambiente físico com a computação. Temos o problema do tempo de simulação, que é o tempo que damos para os objetos físicos se interagirem e tomar o rumo necessário para refletir a realidade.

Existem alguns conceitos como *broad phase*, que é a fase em que os objetos são filtrados para realizarmos depois a *narrow phase* que é onde verificamos se aconteceu alguma colisão entre os objetos.

1.3 Simulador

Com o Physimulation, o aluno da disciplina de física poderá simular comportamentos físicos estudados em sala de aula, por meio de criação de cenários e objetos com diferentes propriedades configuráveis. Exemplos de propriedades são: massa, tamanho, posição, velocidade inicial, coeficiente de atrito e de elasticidade.

Desde rampas, pequenas esferas e plataformas fixas a planetas orbitando em torno de um objeto de grande massa, tanto os alunos quanto o professor poderão utilizar o simulador para combinar tais objetos e observar resultados calculados em exercícios. Além disso, estes cenários podem ser salvos no sistema e carregados posteriormente para nova observação.

1.4 Estrutura da Monografia

Esta monografia está estruturada da seguinte forma: Na seção 2 descrevemos os conceitos e tecnologias estudadas para a realização do trabalho. São apresentadas as bibliotecas utilizadas, o papel de cada uma em nosso projeto e como elas interagem entre si. Nas seções 3 e 4 explicamos dois tópicos comuns e importantes ao realizar simulações físicas: são eles, respectivamente, a discretização da simulação - definição de tempo de simulação, problema do tamanho do passo, granularidade - e o tratamento de colisões - algoritmos para filtragem de elementos, detecção de colisões e quais algoritmos são utilizados no Physimulation.

Já na seção 5 mostramos como foi o passo-a-passo de nosso trabalho, a metodologia utilizada e algumas dificuldades encontradas no caminho. A seguir, na seção 6, apresentamos um dos resultados obtidos: os ambientes físicos de demonstração e as razões por ter começado o trabalho por eles. Logo depois, na seção 7 descrevemos o Physimulation e as etapas para a criação de cenários físicos. Como aplicação do projeto, apresentamos duas integrações com exercícios-programas de disciplinas de introdução a computação na seção 8.

Concluimos o trabalho na seção 9, além de citar possíveis trabalhos futuros e mostrar desafios atuais da computação em simulações físicas que ainda necessitam de estudo e aprimoramento. Ao final, há um roteiro para instalação do ambiente necessário para executar o Physimulation, na seção 10.

2 Plataforma computacional

2.1 Esquema de dependências

Criar uma imagem com as dependências

2.2 Ruby

É a linguagem de programação utilizada no simulador. A sintaxe foi baseada no Pearl e Smalltalk e é uma linguagem interpretada com tipagem dinâmica e forte. Algumas das características da linguagem são:

- Mostrar códigos e alguns itens interessantes do ruby

Explicar sobre gem.

2.3 Simulação com Chipmunk

Chipmunk é uma biblioteca física 2D escrita em C. Esta biblioteca permite criar objetos convexos e segmentos que se interagem em um ambiente físico com algumas propriedades como por exemplo a gravidade.

Para utilizar esta biblioteca é preciso criar as formas dos objetos (Shapes), associá-los a um corpo (Body) e adicioná-los ao ambiente físico (Space). TODO: Explicar melhor cada conceito com códigos

Existe uma versão do Chipmunk portado para a linguagem Ruby.

2.4 Animação com Gosu

A animação é feita com o Gosu, uma biblioteca para desenvolvimento de jogos em 2D em Ruby e C++. Fornece somente as seguintes funcionalidades:

- Criação de uma janela com um laço principal e callbacks
- Criação de textos e imagens 2D
- Som e música em vários formatos
- Tratamento de eventos de entrada de teclado e mouse

TODO: Mostrar exemplos com códigos

2.5 Cenários com Glade

Glade é uma ferramenta que facilita o desenvolvimento de interfaces de usuário em formato de XML. A interface inicial do usuário foi criado com Glade.

TODO: Falar melhor do Glade, visualruby e mostrar exemplos de código

2.6 Exemplo

Mostrar codigos da simulação.

3 Discretização da simulação

3.1 Tempo de simulação

3.2 *Time step problem*

Descrevemos a dificuldade em criar interações físicas reais utilizando a computação.

3.3 Problema de granularidade

Qual deve ser o passo de simulação para ter uma boa velocidade de renderização e precisão física.

4 Colisões

4.1 Algoritmos para detecção

- Geometria computacional
- Alguns algoritmos: BSPTree, QuadTree
- Comparação

4.2 Colisões no Chipmunk

4.3 Efeito de colisões

- ? - Ajuste de colisão vs Correção dos objetos colididos.
- * A posteriori (discrete) versus a priori (continuous) collision detection

5 Atividades realizadas

Falar de como foi o processo do papel até a criação dos demos, interface e integração com ep.

6 Animações produzidas

Apresentação e explicação dos demos.

7 Physimulation

Descrição da interface de criação de cenários e alguns exemplos do que pode ser criado com ele.

8 Introdução à computação com animações

/* Acrescentar algum exercício para o "aluno" fazer. Pode ser
- um dos exercícios programas de MAC2166 que passei para vocês, por exemplo o do "feito estilingue"- o exercício seria "integrar" uma solução com a ferramenta de vocês.

A ideia seria mostrar como um exercício programa de MAC0110 ou MAC2166 pode ser transformado em algo visual (algo parecido com o que o Carlinhos falou)

*/

9 Comentários finais

9.1 Trabalhos futuros

9.2 Desafios

9.3 Conclusão

10 Roteiro de instalação da plataforma

Passos para instalação das bibliotecas: ruby, chipmunk, gosu, chingu, etc.

Falar do warning do próprio chipmunk: `gems/ruby-1.9.3-p286/gems/chipmunk-5.3.4.5/lib/chipmunk.rb:6: Use RbConfig instead of obsolete and deprecated Config`

(Talvez seja um apêndice)

11 Apêndice

O texto a seguir foi extraído do site: <http://bcc.ime.usp.br>.

Material de Apoio ao BCC

A falta de contextualização das disciplinas básicas do BCC tem sido uma queixa recorrente do alunos nas reuniões entre alunos e professores, no Encontro do BCC de 2010 e também no processo de avaliação semestral que é realizado pelo orientador pedagógico orientador pedagógico da Escola Politécnica (POLI), Giuliano Salcas Olguin.

A fim de motivar os alunos e ilustrar a relação entre ciência da computação e as disciplinas básicas de álgebra, cálculo, estatística, probabilidade e física presentes no currículo do BCC a CoC sugeriu que fossem produzidos documentos ilustrando aplicação de cada uma dessas disciplinas em ciência da computação e vice-versa. Esses documentos têm o objetivo de motivar os alunos do BCC:

1. ilustrando as relações entre as disciplinas básicas do curso e ciência da computação;
2. mostrando aos alunos quais das disciplinas mais avançadas do BCC que fazem uso dos conteúdos das disciplinas básicas;
3. fornecendo aos professores das disciplinas básicas do BCC exemplos de aplicações de suas especialidades em ciência da computação, que, eventualmente, podem ser mencionados em aulas ou ser temas de trabalhos.

Esses documentos poderão também ser usados pelas disciplinas de Introdução à Ciência da Computação que são oferecidas pelo DCC para várias unidades da USP. Nestas disciplinas, frequentemente, os chamados exercícios programas ilustram aplicações de métodos computacionais na solução de problemas em genômica, física, economia, etc. Por exemplo, na última edição da disciplina

MAC2166 Introdução à Ciência da Computação para Engenharia

podemos ver um exercício programa em que é simulada a "trajetória de livre de retorno" de uma nave sob a ação gravitacional da Terra e da Lua em <http://www.ime.usp.br/~mac2166/ep3/>. Já um exercício programa com aplicação em genômica pode ser visto em <http://www.ime.usp.br/~mac2166/ep4/>.

Além de uma maior integração do curso este projeto pretende propor possíveis mudanças na grade curricular do BCC. Para isto pretendemos realizar uma pesquisa com o egresso do BCC e uma pesquisa das grades curriculares dos cursos de computação pelo mundo.

Parte II

Parte subjetiva

Nesta seção descreveremos a relação entre nosso projeto e a experiência adquirida no BCC.

(TODO pensei em fazer separado mas agora que escrevi tenho impressão que não vai mudar muita coisa)

12 Alberto Ueda

Entregar este projeto como trabalho de formatura e disponibilizar seu código para os alunos do BCC foram duas das experiências mais gratificantes que já tive. Isto pois acredito que tal conteúdo poderá ser utilizado pelas próximas turmas do BCC como incentivo ao aprendizado da matéria de física. Além disso, tanto alunos do próprio Instituto de Física quanto da Engenharia Politécnica também poderão se interessar pelo conteúdo: o primeiro grupo (FIS) pela animação de fenômenos físicos estudados e o segundo (Poli) tanto pela animação quanto pela simulação de tais fenômenos.

Mas, ao mesmo tempo, por ser um trabalho que levou meses, certas dificuldades foram encontradas pelo caminho. Tivemos que tomar decisões às vezes frustrantes, porém necessárias.

12.1 Desafios e frustrações encontrados

Inicialmente, nossa motivação era entregar um sistema que utilize recursos do Wii Remote (TODO ref TODO link da caneta) e que o professor pudesse utilizá-lo em sala de aula para realizar suas simulações e animações. Porém, chegamos a conclusão que esta tecnologia aumentaria consideravelmente o nível de complexidade de nosso trabalho e não tínhamos garantia de que utilizá-la acrescentaria da mesma forma ao resultado final. Assim descartamos esta possibilidade.

Como utilizamos algumas bibliotecas de terceiros em nosso projeto, tivemos que entender obrigatoriamente como eram feitas as principais chamadas de métodos destas bibliotecas, principalmente o Chipmunk e o Gosu. Um detalhe interessante que ocorreu no segundo mês de trabalho foi a necessidade de mudar o código da biblioteca (TODO citar) e recompilá-la para que uma função simples de mensagem para o usuário funcionasse (TODO conferir método). Uma semana

depois, utilizando uma versão mais nova da biblioteca, descobrimos que nossa alteração não era mais necessária, pois já havia sido feita pelos próprios programadores na mudança de versão.

Além disso, utilizamos um binding (TODO envoltório?) da versão original do Chipmunk. Isto trazia duas dificuldades para nós: 1) o código original (em C++) sempre estava com uma versão mais recente e provia (TODO conferir) mais métodos; e 2) nem sempre o que víamos na documentação oficial possuía correspondente em nosso binding.

Por último, um desafio que tivemos foi encontrar um professor de física disponível para nos auxiliar na elaboração do protótipo do sistema. Ficamos muito felizes quando após algumas semanas o bacharel em física e aluno do BCC João Kerr veio a uma de nossas reuniões, a convite do professor Coelho.

12.2 Disciplinas mais relevantes

- MAC0110 Introdução à Computação : Embora já tivesse contato com programação no ensino técnico, foi nesta disciplina que passei a conhecer e utilizar boas práticas de programação. Além disso, estudei algoritmos famosos e interessantes (por exemplo os de ordenação) que estimularam-me para as disciplinas que viriam a seguir.
- FAP0126 Física I : É a grande motivação deste trabalho. Os conceitos aprendidos nesta disciplina estão por todo nosso código e nas simulações produzidas. Com o Physimulation, tentamos unir o que vimos nesta disciplina com a computação.
- MAC0122 Princípios de Desenvolvimento de Algoritmos : O maior contato com algoritmos, dos mais simples e elegantes aos mais complexos, foi fundamental para minha formação. Primeiro porque me desafiou em certos momentos - e consequentemente me deu coragem para analisar ou implementar futuros algoritmos - e em segundo lugar pois deu-me a confiança de que gostaria de seguir carreira em computação.
- MAC0211 Laboratório de Programação I : Esta disciplina foi interessante por dois motivos: pelo estímulo ao trabalho em equipe e por nos apresentar conceitos e ferramentas relacionadas a qualidade de software, como o Doxygen para documentação de código. Foi nesta disciplina que aprendi o que era um Makefile!

- MAC0323 Estruturas de Dados : Essencial para minha formação como cientista da computação. As estruturas aprendidas nesta disciplina - como listas ligadas e árvores - são muito comuns na programação, mesmo no mercado. Possuem vantagens e desvantagens entre si e o conhecimento de suas propriedades assim como os algoritmos adequados para manipulá-las foram muito importantes para mim.
- MAC0420 Introdução a Computação Gráfica : Outro forte motivador para nosso trabalho. Nesta disciplina tivemos como exercício-programa a simulação de um jogo de bilhar em três dimensões. Foi uma das experiências mais gratificantes do meu BCC, pois minha dupla e eu aplicamos física em um código simples em C com algumas bibliotecas gráficas e de repente tínhamos uma simulação razoável do que ocorre na vida real. Foi quando percebi que com poucos conceitos de física podíamos reproduzir muitos fenômenos naturais, como colisões e dissipação de energia. Percebi também o quanto estes resultados me motivavam a estudar mais, tanto física quanto computação.
- FAP0137 Física II : Os tópicos desta disciplina não foram o foco deste projeto, mas foram grandes motivadores para nosso trabalho. Assim como em Física I, houve pouca contextualização do que foi estudado com o curso de computação. No futuro, temas como relatividade restrita poderão se tornar bem mais simples de se entender por meio de animações criadas pelo próprio usuário, utilizando nosso simulador.
- MAC0332 Engenharia de Software : na área de computação, um dos conceitos mais recorrentes em qualquer projeto de longo prazo é ciclo de vida de um *software*. Este era o tópico mais discutido na disciplina, tornando-a fundamental para o aluno de computação. Outro aspecto a destacar é a prática de trabalho em equipe.
- MAC0338 Análise de Algoritmos: difícil descrever em poucas linhas o quanto esta disciplina é importante para o aluno de computação. Além do desempenho ser uma preocupação constante e necessária a qualquer programador, o aprendizado nesta disciplina é uma das minhas bases sólidas como desenvolvedor. É uma das matérias que quero aprofundar meus conhecimentos durante o mestrado.
- MAC0446 Princípios de Interação Homem-computador : uma das disciplinas mais legais para o aluno que está preocupado com a usabilidade de seu sistema. Os conceitos aprendidos estão por todo o trabalho, assim como em outros projetos que participei ou fui responsável.

12.3 Estudos futuros

Sem dúvida os tópicos de estudo mais importantes para a continuação deste trabalho são as disciplinas de Física I e II para o BCC. Quanto maior o conhecimento das leis e forças físicas presentes no mundo real, melhor serão as simulações e consequentemente as animações geradas.

Em segundo lugar, seria interessante uma análise de qual das alternativas a seguir tem uma melhor relação custo-benefício, visando a atualização do projeto com a versão mais nova do Chipmunk: A) migrar nosso projeto de Ruby para C++ e usar diretamente a versão original do Chipmunk, sem bindings; ou B) atualizar o binding em Ruby adicionando os métodos e funcionalidades da versão mais recente em C++.

Por último, mas não menos importante, um estudo de paradigmas que proporcionem mais usabilidade ao usuário, substituindo o preenchimento obrigatório de formulários para criação de objetos físicos. Ex: *drag-and-drop* do mouse para "arrastar" as formas geométricas, fornecendo os valores de massa, coeficientes de elasticidade e atrito *a posteriori* (após o objeto já estar na tela).

13 Rafael Miyagawa