

Instructions for Authors of SBC Conferences

Papers and Abstracts

Luciana P. Nedel¹, Rafael H. Bordini², Flávio Rech Wagner¹, Jomi F. Hübner³

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

²Department of Computer Science – University of Durham
Durham, U.K.

³Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brazil

{nedel,flavio}@inf.ufrgs.br, R.Bordini@durham.ac.uk, jomi@inf.furb.br

Abstract.

Resumo. *Aqui vai o resumo*

1. Introdução

Aqui vai a introdução.

2. Simulação Computacional

Segundo [Durán 2018] a simulação computacional é uma técnica amplamente utilizada em diversas áreas, como por exemplo no estudo de sistemas complexos que por diversos fatores, não podem ser facilmente reproduzidos.

A simulação permite avaliar diversos cenários de uma pesquisa, apenas alterando suas variáveis de controle, reduzindo assim os custos e os riscos de um experimento físico. Conforme [Marschner and Shirley 2018], essa técnica pode ser aplicada em diversas áreas como biologia, engenharia, física, química, entre outras.

Para construir uma simulação, é necessário uma representação matemática e/ou computacional do comportamento do sistema a ser simulado.

2.1. Formas Geométricas e Movimento

A técnica usada para representar uma forma geométrica computacionalmente dependerá do nível de detalhamento desejado [Pharr et al. 2016]. É possível fazer esta representação de diversas maneiras, por exemplo: forma vetorial, paramétrica ou de forma poligonal. Neste trabalho, adota-se a representação vetorial, com coordenadas cartesianas tridimensionais.

Para fins de simplificação, este trabalho limita-se a modelar objetos na forma de esferas e caixas. Esferas são modeladas com um ponto (centro da esfera) e um raio, enquanto caixas são modeladas ????. <ROGER: REVISAR ESTA SEÇÃO>

As formas geométricas simuladas são estáticas, quando estão fixadas em alguma posição no ambiente, ou dinâmicas, quando estão sujeitas a movimentação provocada por alguma força externa. Normalmente, este movimento é representado pela alteração das coordenadas do objeto dinâmico.

2.2. Colisão e Detecção de Colisão

Uma colisão ocorre quando dois ou mais objetos entram em contato, compartilhando algum espaço do ambiente simulado [Ericson 2004]. Segundo [Bourg and Bywalec 2013], a colisão envolve cálculos para determinar as consequências/efeitos desse contato.

Segundo [Bourg and Bywalec 2013] a detecção de colisão é um problema de geometria computacional, que procura determinar *se* e *onde*, dois ou mais objetos colidiram. Pode ser detectada de diversas maneiras, dependendo da natureza dos objetos envolvidos [Foley 1996], por exemplo: 1) se a distância entre os dois objetos está abaixo de uma determinada tolerância; 2) se os objetos estiverem sobrepondo-se em um ou mais pontos.

O processo para detecção de colisão é dividido em fase de previsão e de identificação. A previsão busca objetos que possivelmente estejam colidindo, enquanto a identificação localiza as colisões ocorridas. A separação em duas fases distintas garante um consumo racional do recurso computacional.

Após detectar uma colisão entre dois objetos, torna-se necessário resolver o evento, atualizando a posição da simulação. Em simulações que não requerem realismo, apenas a correção das posições dos objetos é suficiente. Já em simulações mais avançadas, outros métodos podem ser necessários, como a resolução de colisão por impulso discutida em [Bourg and Bywalec 2013] e [Baraff 1992], onde as características físicas dos objetos são considerados para que a reação seja adequada natural.

3. Resultados Obtidos

O principal resultado obtido é uma aplicação, nomeada de Proto3d, que executa simulações de movimento e colisão de corpos rígidos, em duas ou três dimensões dentro do ambiente configurado pelo usuário. A simulação do movimento é baseada na dinâmica newtoniana e a detecção de colisão é implementada em uma estrutura de dados octree, com algoritmos de previsão (Broad Phase) e identificação (Narrow Phase).

Além disso, o Proto3d foi implementado com o objetivo de ser uma ferramenta acessível às pessoas com deficiência visual. Desta forma, os resultados das simulações são apresentados em uma saída de texto, acessível aos leitores de tela disponíveis no mercado, o que permite ao usuário ler a simulação.

3.1. Funcionamento do Proto3d

O Proto3d tem como finalidade simular o movimento de um ou mais objetos em um ambiente controlado. Nesse ambiente, pode-se controlar os seguintes parâmetros: Gravidade; Limites físicos do ambiente; Passo de tempo.

O objetivo do simulador é representar de forma textual a execução da simulação, bem como permitir o congelamento da mesma, em algum ponto, para análise de algum aspecto que o usuário julgar relevante. Como o Proto3d não possui interface gráfica, foi implementado um pequeno teste, com grandezas conhecidas. O resultado da simulação é registrado em um arquivo de log para análise posterior.

No Proto3d, os objetos simulados são chamados de corpos rígidos, que poderão ser estáticos ou dinâmicos. Um corpo dinâmico é um objeto que possui massa, velocidade, orientação, coeficiente de restituição, forma, e outras propriedades que o pesquisador julgar necessárias. Já o corpo estático é um objeto que possui forma, posição e orientação, mas despreza propriedades como massa, restituição e velocidade. Para este estudo, optou-se em restringir o formato dos objetos em esferas e caixas.

O quadro 1 apresenta a visualização da inicialização deste ambiente. (em nota de rodapé) Algumas linhas do log foram omitidas para adequação ao formato de artigo.

Log 1. Inicialização do ambiente

```

1 Tempo atual: 0.00s; Último passo de tempo: 0.00; Gravidade: True
2 Gravidade: 0.00:0.00:-9.81
3 Limites: 0.00:1000000.00,0.00:1000000.00,-10000.00:1000000.00
4
5 Nome: Chão; Pos: 500000.00:500000.00:-5000.00; Forma Geom: Caixa
6 AABB -> X: 0.00:1000000.00; Y: 0.00:1000000.00; Z: -10000.00:0.00
7
8 Nome: Caixa estática; Pos: 35.00:35.00:5.00; Forma Geom: Caixa
9 AABB -> X: 10.00:60.00; Y: 10.00:60.00; Z: 0.00:10.00;
10
11 Nome: Esfera dinâmica; Forma Geo: esfera;
12 Centro: 10.00:10.00:50.00; Raio: 2.50
13 Mass: 5.00; Restitution: 0.25
14 AABB -> X: 7.50:12.50; Y: 7.50:12.50; Z: 47.50:52.50

```

3.1.1. Formato da saída de dados

O Proto3d registra a saída em um arquivo texto com as seguintes informações:

Visão geral do ambiente

Apresenta informações gerais do ambiente de simulação, como força da gravidade, quantidade, de objetos presentes, bem como uma descrição detalhada de cada objeto no ambiente.

Rastreio do movimento do objeto Após o início da simulação o Proto3d registra as informações para rastreio de movimento do objeto dinâmico em questão. O rastreio do movimento é feito a cada passo de tempo (configurado na inicialização) com as seguintes informações:

- T: Informa a quantidade de tempo decorrido desde o início da simulação até este registro do rastreio.
- Dir: Informa a direção (ou trajetória) do objeto simulado conforme a seguinte legenda:
 - S: O objeto está se movendo para o sul;
 - N: o objeto está se movendo para o norte;
 - O: o objeto está se movendo para o oeste;
 - L: o objeto está se movendo para o leste;
 - C: o objeto está se movendo para cima;

- B: o objeto está se movendo para baixo;
- Pos: Informa o lugar onde o objeto está no instante do registro. Todos os objetos são localizados nos eixos X (horizontal), Y (profundidade) e Z (vertical).
- Vel: Informa a velocidade do objeto no instante do registro. A velocidade do objeto é representada em m/s.

3.2. Simulação de queda da esfera

O objetivo desta simulação é analisar a queda livre de uma esfera sobre uma caixa, que está no chão do ambiente. Na inicialização foi adicionado ao ambiente duas caixas estáticas, uma sobre a outra. A primeira caixa representa o chão (piso) da simulação, enquanto que a segunda caixa, com forma de cubo, está estática sobre o piso. As dimensões destes objetos estão descritas na inicialização do ambiente, explicado na seção 4.1 <usar referência>

O teste consiste em posicionar uma esfera verticalmente alinhada com o cubo, mas em uma posição bem alta e deixar esta esfera cair. Com o avanço da simulação, a esfera será atraída em direção ao chão, devido à força da gravidade no ambiente, até que eventualmente colida com o cubo que está sobre o chão. Quando a colisão ocorre, a esfera recebe um impulso com sentido contrário à queda, que faz o objeto quicar algumas vezes.

Este trecho da simulação pode ser verificado na saída acessível, pelo trecho de código apresetnado em 2.

Log 2. Inicialização do ambiente

1	T: 0.20s, Dir: b, Pos: 10.00:10.00:49.61, Vel: 1.96m/s
2	T: 2.60s, Dir: b, Pos: 10.00:10.00:14.29, Vel: 25.51m/s
3	T: 2.80s, Dir: c, Pos: 10.00:10.00:12.50, Vel: 6.87m/s
4	T: 3.00s, Dir: c, Pos: 10.00:10.00:13.87, Vel: 6.87m/s

Linha 1: Primeira iteração da simulação. A força da gravidade começa a atrair a esfera para baixo.

Linha 2: Iteração no tempo 2.60s. A esfera caiu 35,32m em relação à primeira iteração, estando agora com velocidade de 25.51m/s.

Linha 3: Iteração no tempo 2.80s. A esfera colidiu com a caixa estática e picou, mudando de direção e agora está subindo a uma velocidade de 6.87m/s.

Linha 4: Iteração no tempo 3.00s. A esfera ainda está subindo em função da colisão com a caixa.

Quando a esfera para de picar na caixa, uma nova força (32m/s) é aplicada “empurrando” a esfera para a direita até os limites da caixa estática. No momento em que a esfera “sai de cima da caixa”, a gravidade atrai a esfera em direção ao chão. Deste ponto em diante a esfera está sujeita a duas forças, uma vertical (gravidade) e outra horizontal, uma trajetória descendente na diagonal até chegar ao chão, quando o momento vertical cessa novamente.

Este pedaço da simulação pode ser verificado na saída acessível, pelo trecho de código <referencia> Linha 1: Linha 2: Linha 3: Linha 4:

Ao tocar o chão, a esfera pica novamente, mas agora está sob influência de uma força lateral, descreve um movimento diagonal. No momento em que a esfera parar de quicar, será mantido o movimento lateral até que esta saia do ambiente da simulação.

```
/* trecho quando toca o chão (se conseguir pega um log mostrando a esfera picando) */
```

```
/* fim do trecho */
```

4. Considerações finais:

O Proto3d é uma aplicação desenvolvida com a finalidade de simular o movimento de um ou mais corpos sólidos, detectar e resolver as colisões entre estes corpos e ainda, apresentar a simulação em uma saída acessível a pessoas com deficiência visual.

No simulador, tanto o ambiente como os corpos sólidos, possuem parâmetros configuráveis de forma independente. Os parâmetros do ambiente são: gravidade, dimensão, <listar todos> e os parâmetros dos corpos rígidos são: massa, velocidade, orientação, coeficiente de restituição, forma, e outras propriedades <quais outras? listar todas as que o proto3d permite simular>. Destaca-se que neste relato, para fins de simplificação, optou-se em restringir a forma dos objetos em esferas e caixas.

A simulação executada pelo Proto3d atingiu os objetivos estabelecidos. Os resultados de movimento mostrados no registro de saída podem ser validados com cálculos de física newtoniana. As colisões são detectadas e o movimento resultante também têm resultados coerentes com os esperados.

Como trabalhos futuros podem ser listados: 1) ????, 2) ??????, etc...

5. References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in brackets, e.g. [Knuth 1984], [Boulic and Renault 1991], and [Smith and Jones 1999].

The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the subsequent should be indented by 0.5 cm.

Referências

Baraff, D. (1992). *Dynamic simulation of nonpenetrating rigid bodies*. Cornell University.

Boulic, R. and Renault, O. (1991). 3d hierarchies for animation. In Magnenat-Thalmann, N. and Thalmann, D., editors, *New Trends in Animation and Visualization*. John Wiley Sons ltd.

Bourg, D. M. and Bywalec, B. (2013). *Physics for Game Developers: Science, math, and code for realistic effects*. "O'Reilly Media, Inc."

Durán, J. M. (2018). *Computer simulations in science and engineering: Concepts-Practices-Perspectives*. Springer.

- Ericson, C. (2004). *Real-time collision detection*. Crc Press.
- Foley, J. D. (1996). *Computer graphics: principles and practice*, volume 12110. Addison-Wesley Professional.
- Knuth, D. E. (1984). *The Book*. Addison-Wesley, 15th edition.
- Marschner, S. and Shirley, P. (2018). *Fundamentals of computer graphics*. CRC Press.
- Pharr, M., Jakob, W., and Humphreys, G. (2016). *Physically based rendering: From theory to implementation*. Morgan Kaufmann.
- Smith, A. and Jones, B. (1999). On the complexity of computing. In Smith-Jones, A. B., editor, *Advances in Computer Science*, pages 555–566. Publishing Press.