

Trabalho 4: Spatial Hashing

Guilherme Meneghetti Einloft

1. Introdução

Spatial Hashing é uma técnica utilizada que consiste armazenar objetos do cenário em uma tabela hash que representa o espaço da aplicação, particionado em células de um grid. Essa técnica é usada para encontrar mais facilmente a vizinhança de um objeto qualquer, sem que seja necessário percorrer pela lista completa de objetos.

O trabalho consiste em implementar a técnica para acelerar a detecção de colisão entre círculos aleatoriamente posicionados e uma curva com pontos de controle também aleatoriamente posicionados.

2. Implementação

A implementação foi baseada no artigo “A Hash Table Construction Algorithm for Spatial Hashing Based on Linear Memory” e usou apenas de memória estática para implementação. A estratégia usada foi a de armazenar os círculos na tabela hash.

O algoritmo de colisões entre círculos analisa a célula atual do círculo, além das três outras células mais próximas do círculo, devido ao raio do círculo poder ultrapassar a borda da célula. O algoritmo decide entre 4 direções de checagem: noroeste, nordeste, sudoeste e sudeste.

3. Tempo de processamento

Foi medido o tempo de processamento (em segundos) de todo o processo de posicionamento dos círculos. Deve-se levar em consideração que o posicionamento depende também das colisões entre o círculo e a curva, o que é custoso. O número de segmentos de reta de aproximação da curva é de 2048 e o raio de cada círculo é de 2 pixels.

Num. Círculos / Dim. Grid	2048	4096	8192	16384
S/ Tabela Hash	1.072196	2.269321	6.931638	24.072639
4x4	0.899911	2.078195	5.701384	18.851088
8x8	0.897114	2.021679	5.121567	16.426283
16x16	0.883204	1.999623	4.929432	15.954230
32x32	0.891670	2.008207	5.018745	15.676188

Além do tempo de processamento total, foi medido também o tempo de processamento (em microsegundos) das colisões do último círculo a ser posicionado, para termos uma medida melhor do quanto o algoritmo de hashing espacial acelera o processamento.

Num. Círculos / Dim. Grid	2048	4096	8192	16384
S/ Tabela Hash	92.123	185.037	366.347	755.808
4x4	27.782	59.181	95.378	207.599
8x8	6.652	14.026	27.722	53.871
16x16	1.704	3.196	5.981	19.297
32x32	0.682	1.142	1.643	3.376

5. Conclusão

Na maioria dos casos, observou-se uma diminuição no tempo de processamento, tanto do posicionamento de todos os círculos quanto das colisões entre os círculos, de acordo com o crescimento das dimensões do grid. Entretanto, em alguns casos, quando o grid possuía dimensões 32x32, o tempo de processamento total aumentou em relação ao tempo com grid de 16x16, enquanto o tempo de processamento das colisões diminuiu.

Isso pode ocorrer porque, devido à montagem da tabela hash possuir complexidade de tempo $O(W \cdot H)$, a diminuição do tempo de processamento das colisões não compensa o aumento do tempo da montagem da tabela. Ao aumentar o número de círculos, o tempo de processamento das colisões volta a compensar sobre o tempo de montagem da tabela.

6. Referências

POZZER, C. T.; PAHINS, C. A. L.; HELDAL, I. A hash table construction algorithm for spatial hashing based on linear memory. 11 nov. 2014.