

COMPUTAÇÃO GRÁFICA - TRABALHO 2

Willian Henrique Siman de Lima

Julho de 2019

Detalhes da Implementação

As estruturas de dados que armazenam as primitivas utilizadas para desenhar o terreno são as seguintes: *Terrain*, *Mesh*, *Triangle* e *Vertex*. A classe *Terrain* abre a imagem e carrega as coordenadas de cada vértice, instanciando a classe *Mesh*. A classe *Mesh* contém uma lista de *Triangle*'s, que por sua vez contém uma lista de *Vertex*'s.

Ao instanciar um *Triangle*, é computada sua normal calculando o produto externo entre $R - Q$ e $P - Q$, sendo P , Q e R cada vértice do triângulo, declarados em ordem anti-horária. Ao mesmo tempo que os triângulos estão sendo instanciados, suas normais são somadas num *array* separado. Cada posição desse *array* representa um vértice. Ao instanciar todos os triângulos, o conteúdo de cada posição desse *array* é normalizado e guardado num *Vertex* já existente.

Para mostrar os diferentes modos de visualização requisitados, são usadas *flags*, ajustadas antes de inicializar os dados, que ditam como carregar os dados, que *shaders* usar e que configurações do OpenGL usar.

Escolhas feitas para gerar a visualização

Todas as escolhas feitas para o trabalho anterior foram mantidas em sua maioria. Ou seja, a forma como as transformações são aplicadas no terreno e a projeção usada se mantiveram iguais. Uma alteração foi feita nas cores escolhidas. No modo de visualização com polígonos preenchidos, pode-se apertar a tecla C do teclado para alterar a cor do terreno, de forma a indicar a componente normal em cada vértice como sua cor.

A translação do terreno é feita a partir da movimentação da câmera, e não do terreno em si. Logo, transladar o terreno não causará alterações na iluminação. Entretanto, pode-se movimentar a fonte de luz usando as teclas I, J, K e L do teclado. As teclas I e K fazem a fonte de luz subir e descer, enquanto as teclas J e L fazem a fonte de luz rodar ao redor da origem num plano horizontal nos sentidos anti-horário e horário, respectivamente. A posição inicial da fonte de luz é logo acima da câmera.

O tamanho das setas representando as normais nos modos de visualização 3 e 4 equivale a $1/\sqrt{n_triangulos}$. Dessa forma, o tamanho das setas se mantém grande para terrenos pequenos, sem ficar muito pequeno para terrenos maiores, mas sem poluir a *mesh* de triângulos visualmente.

Falhas e dificuldades

Uma falha presente no programa é que ao aumentar a escala do terreno, as setas das normais não se ajustam ao terreno adequadamente, apontando em direções erradas. O mesmo não ocorre na computação das normais usadas para a iluminação, entretanto.

Computar as normais eficientemente se provou uma tarefa complicada. Inicialmente, para computar as normais do maior terreno de exemplo (*crater.pgm*), gastava-se mais de 5 minutos fazendo a computação (potencialmente levando muito mais do que 5 minutos. Entretanto, nunca deixou-se o programa terminar de computar para medir adequadamente o tempo total da computação). Para resolver esse problema, foram usados recursos da linguagem Python, que processam *arrays* de forma muito mais rápida, reduzindo o tempo de espera para menos de 20 segundos.

Outras melhorias que poderiam ser feitas (não se caracterizando como uma falha, mas simplesmente uma nova função/característica a ser adicionada) seriam permitir a troca entre projeção ortogonal e de perspectiva, um *reset* de todos os parâmetros alterados, permitir a rotação apenas da câmera, independente do objeto, e translação do objeto no mundo (ao invés de translação da câmera).