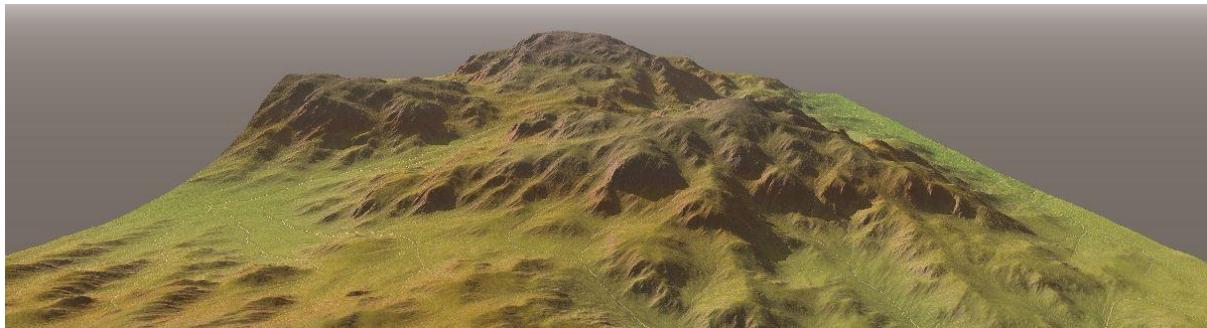


## Trabalho 3 - Terrenos



Desenvolva um programa na Unity (sem usa o terreno que ela oferece nativamente), que permita o carregamento de uma textura que representa o mapa de altura de um terreno. A textura deve ser passada para a GPU onde será feito o *displacement* sobre a malha do terreno. A malha deve ser refinada de acordo com a distância da câmera, garantindo a não existência de *T vertex*, e as normais calculadas através do *geometry shader*. Aplique sobre a malha algum tipo de iluminação que leve em consideração as normais.

A aplicação deve permitir o controle de uma câmera em primeira pessoa para a movimentação sobre espaço, garantindo que a movimentação da mesma seja independente da velocidade do computador. Permitir que o usuário trave o refinamento da malha a qualquer momento, sem perder o refinamento antigo e ainda sim poder movimentar a câmera. Além disso, adicionar opção para visualização da malha em *wireframe*.

### Extras:

- Implementação de uma *Quad Tree* para a remoção de partes da malha que estão fora da visão da câmera.
- Adicionar física sobre a superfície do terreno. Exemplo: interação de objetos, veículos com suspensão sobre o terreno.
- Adicionar detalhes sobre o terreno usando algum tipo de ruído (*Perlin Noise*, *Fractal*, etc).
- Aplicar diferentes texturas sobre o terreno levando em consideração a altura, declive, por exemplo.
- Adição de sombra
- Etc.

### Observações:

- Incluir um arquivo “**README.txt**” contendo informações sobre quais funcionalidades foram implementadas (requisitos e extras).

### Data e Formato de Entrega:

- Data: 27/out/2017.
- No email e no cabeçalho do arquivo, devem conter o nome completo e matrícula do aluno. O arquivo deve ser enviado para [pozzer3@gmail.com](mailto:pozzer3@gmail.com) com subject “**CGA T3**”.

- O programa deve ser enviado em um arquivo compactado **fulano.rar** (fulano = login ou nome do aluno), **sem links para dropbox, drive, etc.** Dentro deste arquivo deve haver um diretório com o mesmo nome do arquivo e dentro deste diretório os arquivos do trabalho.
- **Ex:** o arquivo **pozzer.rar** deve conter um diretório chamado pozzer, e dentro do diretório devem estar os arquivos do trabalho. O arquivo .rar pode ser renomeado para .rar.txt caso o gmail não aceite.

**Critério de Avaliação:**

- **documentação:** descrever no cabeçalho de cada arquivo a ideia geral do código e detalhes específicos de partes que mereçam uma explicação - não comente por exemplo o que faz b++.
- **pontualidade:** Trabalhos não entregues na data não serão avaliados e receberão **nota zero**.
- **legibilidade:** nome de variáveis, estruturação do código.
- **clareza:** facilidade de compreensão - evite códigos complexos e desnecessários. Adote a solução mais simples possível.
- **funcionalidade:** o programa deve satisfazer todos os requisitos. Programas que não compilarem ou que não atenderem a nenhum requisito receberão **nota zero**.

**Você pode discutir estratégias e ajudar o colega na implementação, porém evite passar código fonte. Programas semelhantes terão nota zero.**