Foi desenvolvida uma primeira versão do programa, denominada “Alpha”, para experimentar a viabilidade da plataforma Unity no desenvolvimento do programa proposto. A plataforma providencia ferramentas para criar e renderizar um relevo em um ambiente 3D, e funções para alterar a estrutura do relevo em tempo de execução (Unity).

Para ter uma referência de como a simulação funcionaria, foram desenvolvidos dois algoritmos simples de transformação de matriz. O primeiro algoritmo, “Suavização”, altera a altura cada ponto do terreno para o resultado da média das alturas que o cercam, causando a suavização de partes mais escarpadas do terreno. O segundo algoritmo, “Escavação Lateral”, é uma adaptação do algoritmo anterior. Nesse algoritmo, a média dos valores é feita apenas dos pontos a partir de uma direção específica (por exemplo, norte), e o algoritmo só mantém alterações de altura para baixo, não para cima. O resultado é uma aproximação do efeito do vento sobre areia a partir de uma direção fixa. Combinando os dois algoritmos, é possível ver a formação de dunas.

Um terceiro algoritmo, “Depósito”, consiste em uma adaptação do algoritmo de suavização para tentar simular o assentamento de areia fina. Inicialmente, é realizada a suavização, apenas considerando alterações de altura para baixo. Todas as variações são armazenadas e redistribuídas igualmente para os vizinhos mais baixos. Embora esse algoritmo tenha mais sucesso em manter a quantidade de matéria do terreno o mais constante possível, ele é o mais lento dos três, e não muito discernível da suavização, sendo deixado de lado.

Para os dois outros algoritmos, foram implementadas duas parametrizações para cada: alcance e intensidade do efeito. O alcance indica qual a área ao redor do ponto do terreno que deve ser considerada ao aplicar a média, e a intensidade é utilizada para reduzir a velocidade na qual as alterações surtem efeito. Verifica-se que a utilização de áreas maiores implica em uma suavização mais intensa, com inclinações menos acentuadas e transições mais prolongadas.

Na pesquisa foi encontrado uma referência para algoritmos de simulação de erosão, assim como maneiras de otimizar um algoritmo (por exemplo, com uso de vizinhança Von Neumann). Essas otimizações passarão a ser consideradas no desenvolvimento do programa.

@@ PARA MONOGRAFIA @@

REPRESENTAÇÃO COMPUTACIONAL DO RELEVO

Antes de iniciar o desenvolvimento da simulação, é necessário entender as técnicas utilizadas para se armazenar e representar um relevo em uma aplicação computacional. A plataforma Unity utiliza mapas de altura (*heightmaps*) como forma de armazenar as informações de objetos do tipo terreno.

Um mapa de altura consiste nos valores de altura do relevo, organizados em uma matriz, de tal forma que essas alturas podem ser mapeadas a suas respectivas coordenadas (x,y). Essa matriz pode ser visualizada como uma imagem bitmap em escala de cinza, onde cada valor de altura corresponde ao valor entre preto e cinza de um pixel na imagem. Essa característica tem a vantagem de oferecer uma visualização compreensível do terreno na forma de bitmap, e possibilita o uso de editores de imagem para criar ou modificar um relevo. (Livro Terrain Engines in C++)

Na plataforma Unity, os objetos de terreno utilizam dados no formato mapa de altura como entrada. O formato aceito é [...], sendo carregado na memória como uma matriz de valores de ponto flutuantes de zero a um. No entanto, é possível carregar os dados de um arquivo de imagem em tempo de execução e construir a matriz a partir das informações. É importante ressaltar que, embora o Unity aceite mapas de resoluções variadas, a matriz deve ter sempre um formato quadrado. (Entretanto, o Unity permite que o objeto de terreno seja distorcido para ter um formato retangular).

ALGORITMOS DE TRANSFORMAÇÃO DE RELEVO

Tendo em vista que, no Unity 3D, os dados de altura de um objeto de terreno são armazenados por meio de uma matriz de valores flutuantes de zero a um, pode-se analisar esta estrutura como uma imagem em escalas de cinza, onde *pixels* completamente pretos correspondem a pontos de altura mínima, e *pixels* completamente brancos correspondem a altura máxima.

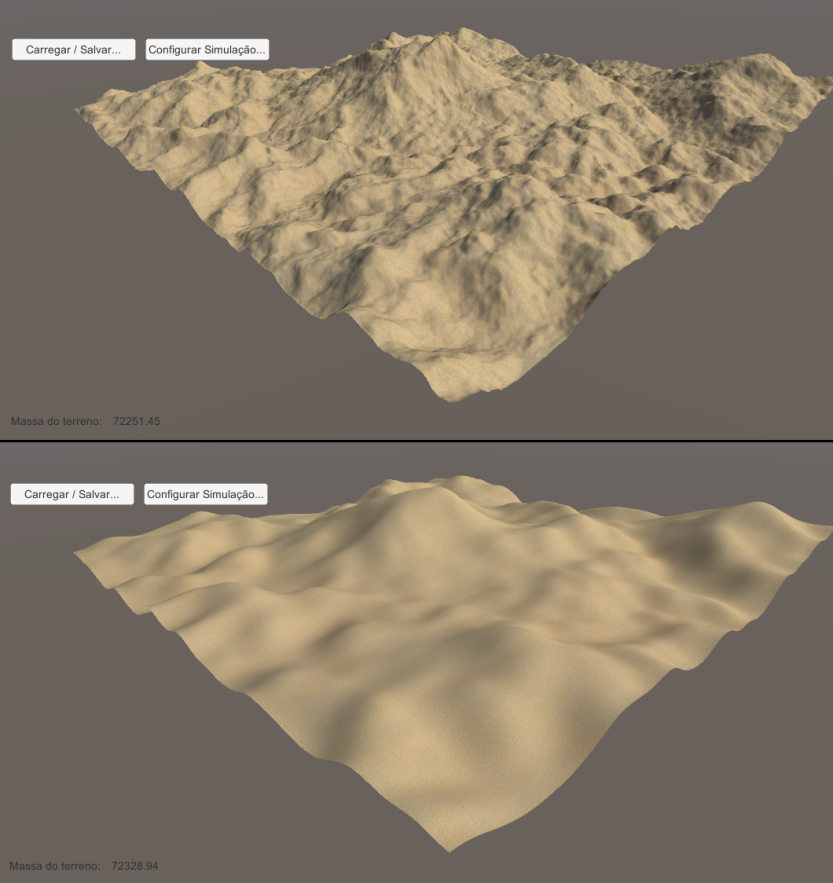
Levando-se esta análise em consideração, podem-se aplicar os conceitos utilizados na filtragem de imagens no desenvolvimento de algoritmos que transformem o relevo. Um dos algoritmos mais simples utilizados em imagens é o *box blur*, onde para cada ponto da matriz é feita uma média aritmética simples dos valores adjacentes, e então alterado o valor central para esta média. Este algoritmo tem o efeito de reduzir o ruído e borrar (*blur*) a imagem como um todo (CHANDEL; GUPTA, 2013, v. 3 ed. 10 p. 198).



- Exemplo de utilização do filtro Box Blur na ferramenta StylePix (Hornil StylePix User Manual)

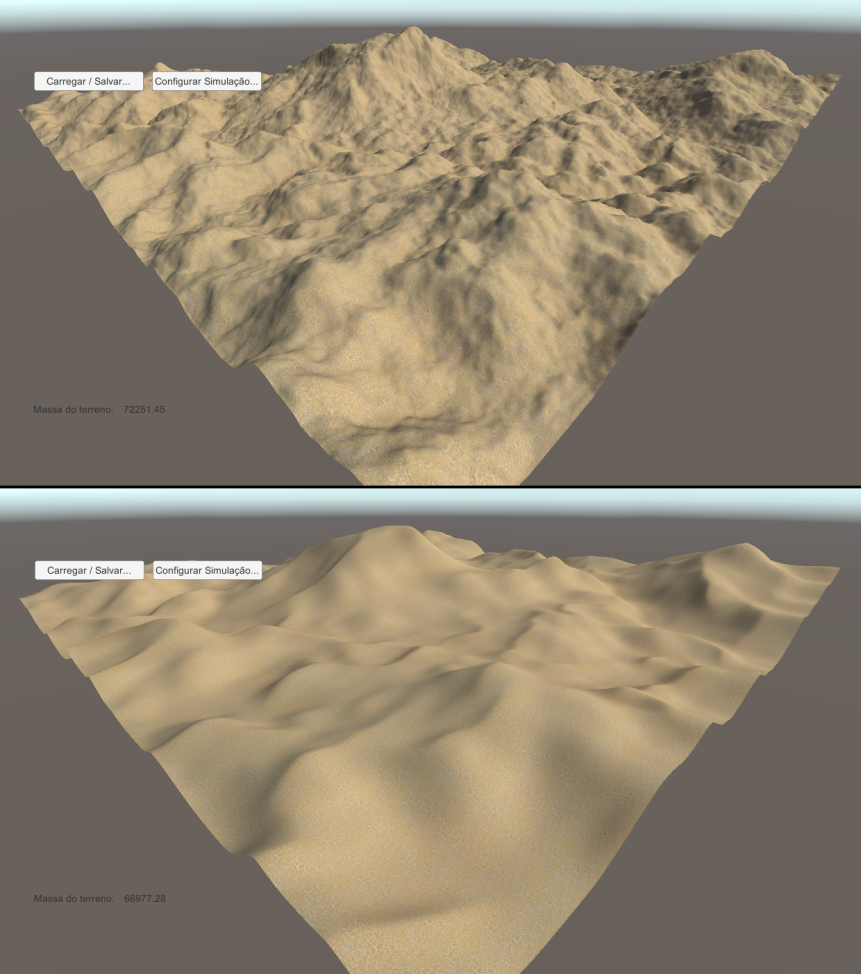
Na primeira aplicação do algoritmo de *box blur* sobre os dados de terreno do Unity, foi considerada inicialmente uma matriz retangular de 3x3 centralizada sobre a célula que será atualizada. As alterações não são feitas imediatamente no terreno em alteração, sendo salvas em uma matriz secundária que substituirá a matriz inicial no término da execução. Desta forma, evita-se que alterações feitas em um ponto afetem alterações nos próximos pontos.

A execução do algoritmo elimina pequenas características do relevo e suaviza inclinações, um resultado que pode ser comparado ao assentamento de areia solta. Foi desenvolvida então a possibilidade de parametrização de certos componentes do algoritmo, inicialmente a área da matriz considerada no cálculo da média, assim como um “fator de transformação”, que permite reduzir a alteração do terreno a cada iteração do algoritmo.



- Resultado da execução do algoritmo box blur sobre um relevo (fonte própria)

Para avaliar as possibilidades que esse formato de algoritmo providencia, foi desenvolvida uma alternativa ao primeiro algoritmo. Ao invés de considerar a matriz de vizinhos completa na aplicação da média, são consideradas apenas as células que se encontram na mesma latitude ou abaixo dela. Além disso, são descartadas alterações que elevem partes do terreno, permitindo apenas o rebaixamento dos pontos de altura do mesmo. O resultado é uma transformação similar ao efeito do vento varrendo uma paisagem arenosa a partir de uma direção fixa. Aplicando-se os dois algoritmos simultaneamente, verifica-se a formação de estruturas similares a dunas.



- Resultado da combinação do box blur com o algoritmo personalizado (fonte própria)

Tais algoritmos podem ser otimizados de diversas maneiras, ao custo de acurácia. Parametrizar matrizes de vizinhança menores [resultado]. Tal otimização é desejável, tendo em vista que o ganho de qualidade que se vê ao utilizar matrizes maiores é negligível para os fins da aplicação proposta. Outras maneiras de se otimizar esses algoritmos podem ser verificadas em Olsen (2004, p. 6) [citar essas maneiras, só que elas estão em formato de lista?]

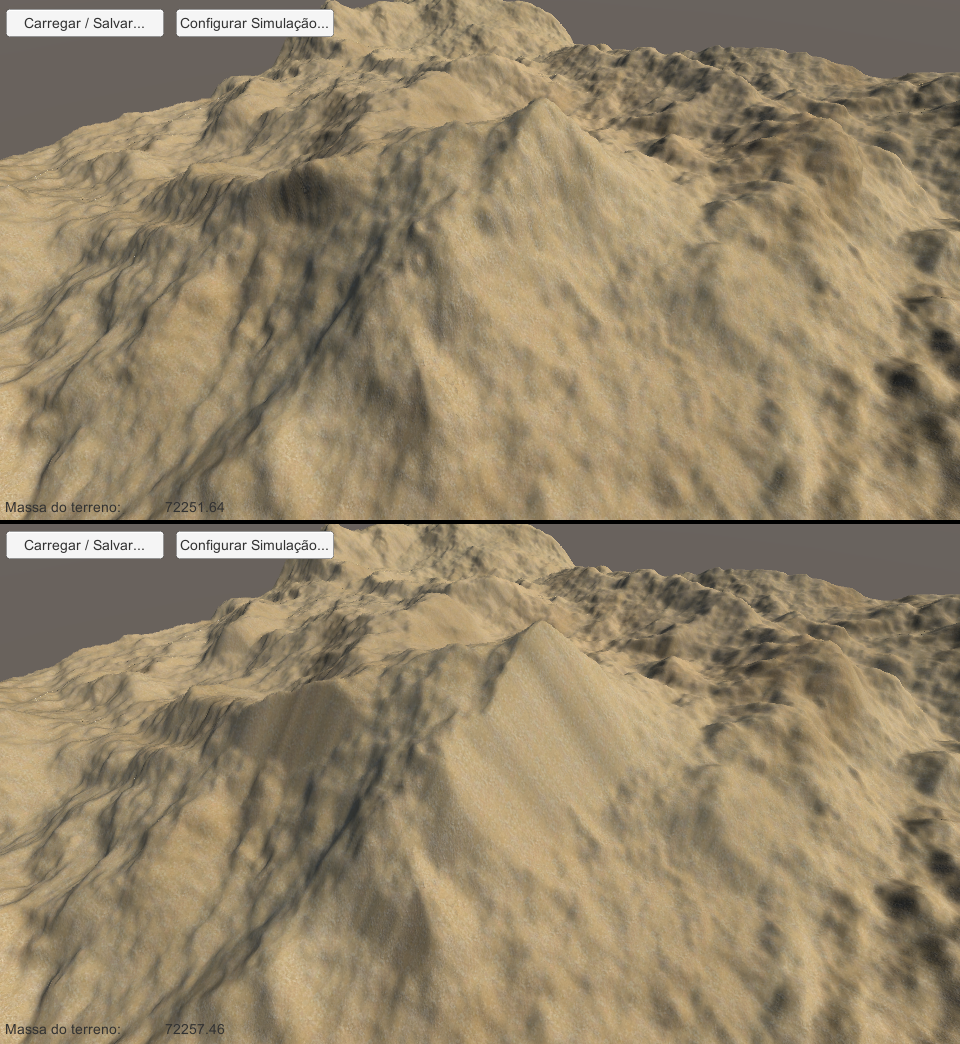
ALGORITMOS DE EROSÃO

Para simular os efeitos da erosão no terreno, foram utilizados os algoritmos detalhados em Olsen (2004, p. 5-6) como base.

O primeiro algoritmo visa representar os efeitos da erosão térmica, que causa a queda de material em encostas e o seu acúmulo na base. O algoritmo atinge esse efeito através da movimentação de material de pontos mais alto para vizinhos mais baixos caso a diferença entre os dois ultrapasse um limite pré-definido. No entanto, como pode haver múltiplos vizinhos que satisfaçam essa condição, é necessário realizar a distribuição do material movimentado de forma proporcional às inclinações correspondentes.

[Algoritmo da erosão térmica]

Onde [T] é a inclinação máxima permitida (representada pela diferença entre duas alturas adjacentes) e [C] é um fator de movimentação de material. Este fator reduz a quantidade movimentada por iteração do algoritmo, deixando a transformação mais devagar, mas também melhorando a qualidade da simulação.



- Resultado da execução do algoritmo de erosão proposto (fonte própria)

O segundo algoritmo simula a erosão a partir de forças hidráulicas, ou seja, o deslocamento de material causado pela precipitação e escoamento da água. A primeira parte do algoritmo descreve uma simulação simplificada da hidrografia do relevo através de um mapa auxiliar de volume de água. Para fins de descrever esta simulação, têm-se as informações de altura da superfície do relevo *h*, volume de água *a*, e altura da superfície da água *s* (sendo *s = h + a*).

A simulação da ação da chuva é realizada através da soma de um valor de precipitação a todos os valores de *a*. Então, é subtraída de todos os valores *h* uma parcela proporcional à quantia de água presente sobre a célula, de acordo com um parâmetro de solubilidade *S*. Finalmente, a água é escoada através de uma fórmula similar à utilizada na erosão térmica, mas que visa nivelar os valores de *s* através da movimentação de água para os vizinhos inferiores.

[Fórmula de distribuição de água]

Após o escoamento, ocorre a evaporação/absorção da água, através da subtração de um percentual de cada valor *a*. Durante esta operação, é retornado à *h* o material contido na quantidade de água removida, seguindo a mesma proporção utilizada na remoção do material.

[Exemplo erosão hidráulica]

A aplicação destes algoritmos mostrou resultados satisfatórios, mas não provou ser suficiente para uma simulação mais representativa da realidade. Propõe-se então uma expansão destes algoritmos, através da inclusão de parâmetros que não foram considerados em sua concepção inicial.

INCLUSÃO DA CAMADA DE ROCHA

Para aproximar a simulação de uma situação real, passa a se considerar um segundo objeto de terreno com as mesmas dimensões do terreno original, mas com uma matriz de alturas distintas. Este novo mapa representará a estrutura da camada de rocha presente abaixo do solo. Tendo as alturas h(solo) e h(rocha), pode se calcular uma quantidade de solo presente em cada célula através de h(rocha) – h(solo). Nota-se que a altura do solo em qualquer célula deve ser igual ou superior à altura da rocha, senão haveria a possibilidade de existirem células contendo quantias negativas de solo. Essa regra deverá ser levada em consideração na adaptação dos algoritmos a seguir.

Tendo em vista o algoritmo de erosão térmica desenvolvido anteriormente, é necessário reformular o cálculo de distribuição do solo entre as células vizinhas que estão propensas a receber material. A quantia movimentada para cada vizinho passa a ser multiplicada por um fator limitante [L], que em sua vez é obtido usando a seguinte fórmula:

[Fórmula do fator limitante (quantia de solo / total a ser movido)]

Quanto ao algoritmo de erosão hidráulica, é realizada uma simples verificação na conversão de solo em sedimento: Quando a quantidade a ser convertida for maior do que a quantidade de solo presente na célula, é convertida apenas esta quantidade. Nota-se que, como o algoritmo atual assume uma concentração uniforme de sedimento por toda a água, equivalente a quantidade máxima de solo convertida, que essa limitação causará um lento aumento da quantidade final de solo na paisagem.