



**ÁREA DEPARTAMENTAL DE ENGENHARIA DE
ELECTRÓNICA E TELECOMUNICAÇÕES E DE COMPUTADORES**

**Licenciatura em Engenharia
Informática e Multimédia**

Interação em Ambientes Virtuais

Trabalho Prático 1 2021SV

Docente: Eng. Arnaldo Abrantes

Aluno:

José Siopa, A46338

2 de maio de 2021

ÍNDICE

1	INTRODUÇÃO	1
2	DESENVOLVIMENTO	3
2.1	Implementação base	3
2.2	Implementações adicionais	5
3	CONCLUSÃO	7

1 Introdução

Como primeiro trabalho prático e, utilizando como base o código desenvolvido nas aulas, foi pedido a realização de um ambiente tipo Minecraft. O trabalho foca-se na parte de geração procedimental de terreno utilizando o ruído de Perlin para criar uma superfície aleatória, mas com suavidade, simulando o relevo do mundo real.

Este terreno é criado recorrendo a *voxels*, cujos quais são gerados dinamicamente e são totalmente manipuláveis.

2 Desenvolvimento

2.1 Implementação base

Depois de desenvolvido o código nas aulas, foi necessário corrigir alguns problemas existentes na geração do mundo, bem como implementar novas funcionalidades para enriquecer o trabalho.

O primeiro problema que apareceu foi que os *chunks* já criados não estavam a ser redesenhados quando o jogador voltava a aproximar-se deles. Assim, foi necessário criar uma lista auxiliar com apenas os *chunks* que estão atualmente ativos e outra lista com todos os *chunks* já criados. Desta maneira, não é necessário estar a recriar *chunks*, apenas se altera a visibilidade do *chunk*, o que irá otimizar um pouco a geração do mundo.

Estando este problema resolvido, passou-se à manipulação dos valores do mundo, para obter um resultado que seja o mais aproximado possível ao do jogo Minecraft. Através do uso da função de movimento browniano fractal obtém-se um terreno realista com diferentes declives. Dependendo da altura resultante da função, é possível criar biomas, por exemplo, lagos ou montanhas. Para futura implementação, como se pode visualizar na Figura 1, pode-se colocar blocos do tipo água.

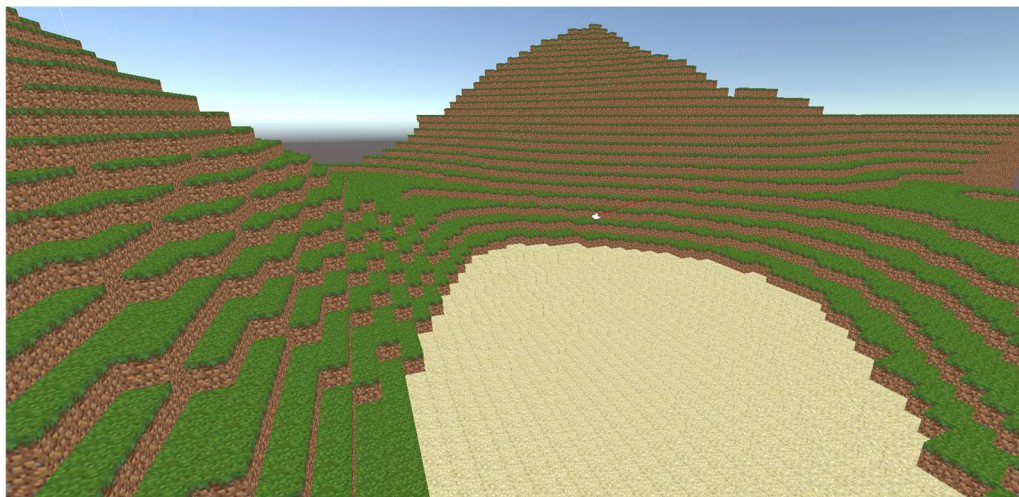


Figura 1 - Superfície do terreno

Relativamente ao terreno debaixo da superfície, utilizou-se a mesma função de movimento browniano fractal em 3D, que consiste na média da função anteriormente usada para a superfície, para todos os pares de coordenadas. Assim, é possível gerar cavernas aleatoriamente pelo subsolo (Figura 2).

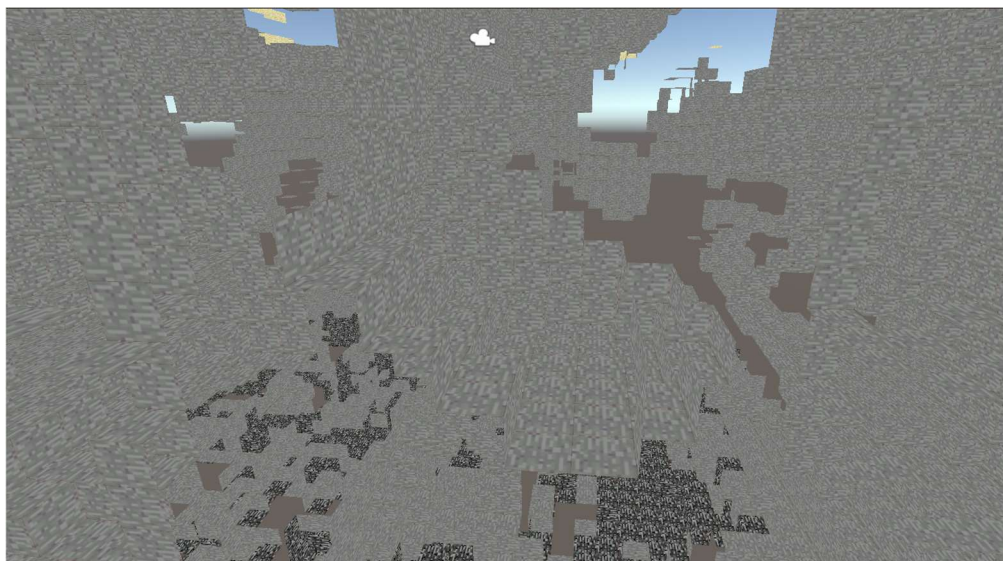


Figura 2 - Subsolo do terreno

A geração recursiva dos *chunks* está atualmente a utilizar a vizinhança de von Neumann que, apesar de parecer mais otimizada comparativamente com a vizinhança de Moore por carregar menos *chunks*, irá ter que renderizar mais faces.

Foi também utilizado um atlas diferente daquele utilizado nas aulas, com ainda mais blocos. O código teve de ser adaptado pois este atlas tem uma largura superior.



Figura 3 - Atlas utilizado no mundo

2.2 Implementações adicionais

Como implementações adicionais, foram feitas alterações aos blocos do terreno. Como se pode observar na Figura 1, a partir de uma certa altura para baixo, aparecem blocos de areia. A ideia seria criar lagos aleatoriamente pelo mundo preenchidos com água.

Nas primeiras camadas do mundo, foi criado o bloco de *bedrock*, que impede que o jogador caia para o infinito. Este bloco, tal como no Minecraft, é impossível de ser quebrado. Além disso, nenhum *chunk* é criado abaixo da camada de *bedrock*.

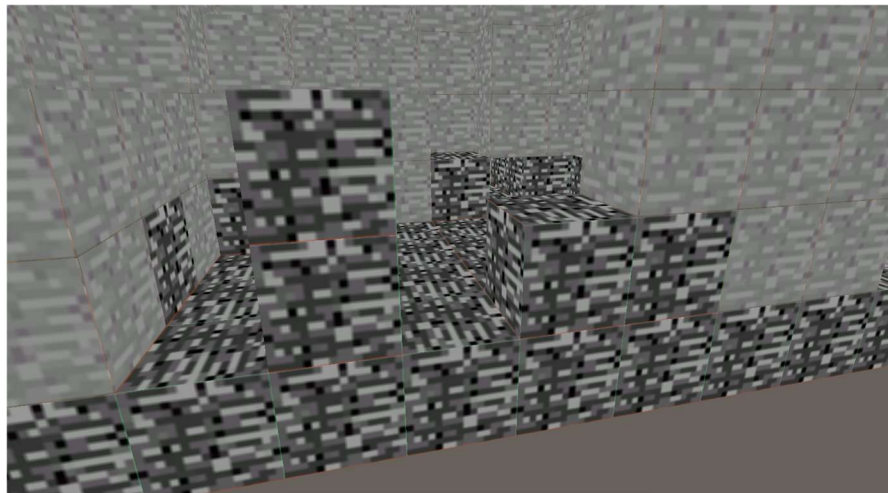


Figura 4 - Primeiras camadas do mundo

Relativamente à interação do jogador com o mundo, este pode colocar diferentes tipos de blocos, com ajuda de uma interface que permite a distinção de qual bloco irá colocar (Figura 5). Também é possível quebrar todos os blocos já existentes e os colocados pelo jogador, com exceção do bloco de *bedrock*.

Para se saber qual o bloco que o jogador está a olhar, foi criado um *loop* que em cada iteração irá incrementar um valor fixo ao vetor que aponta na mesma direção que a camera até atingir a distância máxima de alcance. Se durante alguma dessas iterações as coordenadas do vetor colidirem com um bloco que não seja do tipo ar, irá ser colocado um bloco invisível na mesma posição que o bloco já existente, sobrepondo-o. Caso o jogador dê um clique com o botão esquerdo do rato, o tipo do bloco já existente é mudado para ar, passa a não ser sólido e o *chunk* é redesenhado. Caso dê um clique com o botão direito do rato, o bloco que está atualmente selecionado é colocado na posição da iteração anterior do bloco que está a ser olhado pelo jogador.



Figura 5 - Exemplo da colocação de blocos

3 Conclusão

Estando concluído o trabalho, sente-se que ainda é possível otimizar ainda mais o jogo. Ao invés de se utilizar corrotinas, poder-se-ia utilizar tarefas para gerar e desenhar o mundo. As faces opostas ao jogador também não necessitavam de ser renderizadas.

Se houvesse mais tempo, também seria interessante implementar árvores geradas proceduralmente e a criação de biomas e minérios.

Outra implementação que queria implementar, seria a adição de blocos transparentes, sem colisão e com movimento, tais como água, em lagos e oceanos, e lava, em algumas grutas.