

Docente: Arnaldo Abrantes  
Alunos: Miguel Silvestre nº45101 | Pedro Dias nº 45170  
  
  
Trabalho efetuado no âmbito da cadeira letiva de IAV do Curso de Engenharia Informática e Multimédia.

Trabalho Prático 1

Interação em Ambientes Virtuais | Semestre Verão 21/22

Índice

[Indice de Figuras 2](#_Toc101466349)

[1-Introdução 3](#_Toc101466350)

[2-Desenvolvimento 4](#_Toc101466351)

[2.1 – Construção do Mundo 4](#_Toc101466352)

[2.1.1 – Texturas 4](#_Toc101466353)

[2.1.2 – Ruído de Perlin 5](#_Toc101466354)

[2.1.3 – Parâmetros 5](#_Toc101466355)

[2.2 – Funcionalidades Adicionais 6](#_Toc101466356)

[3-Conclusão 6](#_Toc101466357)

[4-Bibliografia 7](#_Toc101466358)

# Indice de Figuras

[Figura 1 - Atlas 5](file:///D:\GitHub\IAV-TP1\IAV_TP1\Relatorio\relatorio_iav_tp1.docx#_Toc101466360)

# 1-Introdução

Este trabalho visa a criação de um mundo num ambiente do tipo Minecraft em Unity.

Tendo como base o trabalho efetuado ao longo das aulas, onde se criou, com a utilização de técnicas procedimentais, mundos baseados em voxels, a geometria 3D do mundo (tris, quads, blocks, chunks), e se utilizou o ruído de Perlin para criar o terreno de forma processual, iremos construir a nossa própria versão acrescentando funcionalidades adicionais.

# 2-Desenvolvimento

## 2.1 – Construção do Mundo

De modo a ser possível termos um mundo infinito, é necessário este estar otimizado, de modo a diminuir a carga computacional. Assim, neste projeto, um mundo é criado continuamente em torno do agente à medida que este se movimenta e por consequinte, parte do mundo é “destruído” se este se afastar consideravelmente.

Como dito anteriormente, este projeto visa a criação de um mundo Minecraft, ou seja um mundo baseado em cubos. Neste ambiente, existem faces que não são visíveis ao jogador, mas que ao serem criadas e renderizadas tornam a execução mais lenta. Assim, de forma a reduzir o número de triângulos, vertices, etc, e melhorarmos o desempenho, não processámos estas faces.

### 2.1.1 – Texturas

Para ser possível ao jogador distinguir os diferentes tipos de cubos, a cada um foi adicionado uma textura diferente, permitindo diferenciar a relva, a pedra, a terra, o ar, o diamante e o ouro. Isto foi realizado recorrendo a um atlas. Este, apesar de ser o mesmo utilizado nas aulas, tem texturas adicionais, nomeadamente diamante na posição (2,12) e ouro em (0, 13).



Figura - Atlas

### 2.1.2 – Ruído de Perlin

O ruído de Perlin é uma técnica muito usada em jogos e utiliza uma série de números parcialmente aleatórios com o objetivo de imitar objetos naturais, como o sol, nuvens, animações, terrenos, entre outros. Permite o controlo de elementos de pequena e grande escala e utilzamos este ruído de Perlin de forma a nivelar o nosso terreno. Quando maior for a sua regularização (smooth), mais nívelado fica o terreno.

### 2.1.3 - Algoritmo de Flood Fill

De modo a ser possível construir os diferentes tipos de blocos (chunks), utilizamos o algoritmo de Flood Fill que os desenha recursivamente. Em termos gerais, este algoritmo determina a àrea conectada a um nó dado em um vetor multi-dimensional de nós. Isto possibilita-nos a gerar mais do que um bloco do mesmo tipo, criando assim blocos vizinhos.

### 2.1.4 – Parâmetros

TODO FALAR AQUI OS PARAMETROS BLA BLA BLA + DESEMPENHO DE FPS, TRIANGULOS, VERTICES BLA BLA BLA TODO

## 2.2 – Funcionalidades Adicionais

TODO ESPERAR LED TODO

TODO ALGORITMO FLOODING ACIMA? TODO

# 3-Conclusão

Este trabalho foi interessante, pois toca num jogo bastante popular e com muito sucesso do nosso tempo atual.

A otimização é deveras importante, caso contrário não seria possível ao jogo fluir naturalmente, devido a carga computacional, congelando o ecrã por alguns segundos enquanto novas partes do mundo são criadas. Apesar do terreno ter sido criado com sucesso, este não está perfeito e existem melhorias a ser feitas, nomeadamente por vezes, alguns blocos por baixo das grutas não são construídos, sendo possível ao jogador cair e não conseguir proceder o jogo.

Neste tipo de jogos, existe um número incontável de funcionalidades adicionais que podem ser acrescentadas, o limite é a nossa imaginação. No entanto, salienta-se a possibilidade de o jogador conseguir destruir vários tipos de blocos. Infelizmente, o tempo não foi suficiente para conseguirmos realizar esta interação.

Apesar da existência de alguns problemas e haver espaço para melhoramento, consideramos que o trabalho cumpre os requisitos e foi realizado com sucesso.

# 4-Bibliografia

1. Vídeos das aulas no Youtube fornecidos pelo docente – Arnaldo Abrantes.
2. <https://en.wikipedia.org/wiki/Perlin_noise>
3. https://en.wikipedia.org/wiki/Flood\_fill