

# Computação Gráfica

## Relatório

Nome: Orlando Daniel Silva Cruz

Nº: 1502834

### LineMP

Para o desenvolvimento do ponto médio tive por base o algoritmo de Bresenham, efetuando algumas alterações tornando-o funcional para qualquer tipo de pontos (negativos ou positivos). Nesse sentido algumas considerações foram tomadas, desde a inclinação da reta, adaptando o ponto medio para ângulos mais ou menos acentuados, em que nesse caso o pixel resultante pode estar à esquerda/direita ou cima/baixo, bem como a origem e final da reta, o que pode significar que se deve incrementar ou decrementar x ou y.

### Display

O display apresentado possui 21 pixels formados por quadrados. Cada quadrado possui uma posição específica no plano sendo que essa posição será útil na detecção de interseções. Na criação do display os quadrados são criados com dois tipos de materiais, apresentando cada um uma cor diferente no display. A obtenção de um padrão xadrez é feito considerando a posição que será atribuída ao quadrado, é, portanto, necessário considerar posições em x como em y. O ciclo começa por atribuir um material ao quadrado se a sua posição em y for um número par ou ímpar, alternando depois a medida que vai percorrendo o eixo x. Sempre que se muda de linha (incrementa-se y), o material no eixo y vai alternando, influenciando todo o eixo x.

Os eixos são apresentados por duas linhas construídas por pontos iniciais e finais sendo o seu comprimento absoluto =  $21/2$ .

A utilização do OrbitControls é apresentada com limitações em relação ao deslocamento, sendo apenas possível rodar em torno da origem e aumentar o Zoom.

### Posições do rato e interseções

Para se conhecer em que coordenada se encontra o rato é necessário conhecer as dimensões de tela na qual o rato se pode deslocar. Para obter a posição correta do rato utilize as recomendações da documentação da biblioteca Three.js.

Para se obter a coordenada de cada pixel, foi necessário a utilização do raycaster. Neste caso, à medida que o rato se vai movimentando é verificado se está dentro ou fora do plano. Se estiver dentro do plano, então estará em cima de algum pixel. Os pixels são capturados através de intersectObjects, e cada objeto capturado é verificado pelo seu nome. Se o seu nome é compatível com o que está associado então é obtida a posição de x e y do objeto efetuando o seu log. Para evitar que pequenas movimentações sobre o mesmo objeto resultem no seu log, é definida duas variáveis que retém a posição x e y do último pixel detetado, assim no caso de o objeto ser o mesmo não existe log da sua posição.

## **Teclas e posições**

Cada vez que a tecla é pressionada é realizada a sua detecção por meio de um evento. Caso a tecla seja a tecla x é necessário realizar as seguintes verificações:

- O rato está posicionado dentro ou fora do plano?
- O ponto é inicial ou final?

Para se saber se o rato está ou não posicionado no plano verifica-se se o rato está posicionado em cima de um pixel. Esta verificação é realizada sempre que existe um movimento do rato.

O ponto é considerado inicial ou final através de um contador. O contador inicia a 0, sendo que vai incrementando sempre que se pressiona a tecla x assegurando que a posição do rato está posicionada em cima de um pixel. O contador é restaurado a zero depois de um ponto final ser escolhido pelo utilizador, permitindo escolher diversos pontos.

## **Ladrilhos e retas**

Para se obter uma reta é necessário que utilizador efetue a escolha de dois pontos no plano. Sempre que um ponto é escolhido, o material do objeto é substituído por um novo, substituído a cor original por vermelho.

A escolha de dois pontos consecutivos resulta na criação de uma reta e de pixels amarelos (cubos) que correspondem ao conjunto de pontos calculados pelo LineMP. Nos pixels amarelos (cubos), é aplicada uma transparência de forma a ser possível visualizar a reta correspondente. Os objetos (retas e cubos) criados possuem um nome específico para seja mais fácil a sua eliminação.

## **Limpeza e menu**

É possível efetuar a limpeza do display utilizando a tecla backspace ou pelo menu. A limpeza é efetuada tendo em consideração quais são os objetos que interessam limpar. Com esse objetivo obtém-se todos os objetos presentes em cena em que o seu nome corresponde ou a retas ou a pixels. Caso existam, estes são eliminados através de um ciclo até que já não existam mais objetos do mesmo nome. Após a limpeza o display é restaurado às suas características originais.

O menu foi criado apenas por curiosidade. A maior utilidade está em recentrar o plano que durante o desenvolvimento foi bastante útil. Para recentrar o plano basta fazer reset ao controls, que por sua vez volta a ter os valores originais.