

Computação Gráfica

Relatório - Trabalho 1

2022-2023

Grupo CG69

- Nº 56278 - Henrique Catarino
- Nº 56338 - Miguel Nunes
- Nº 56303 - Tomás Piteira

PONTO I:

Para representar objetos tridimensionais em superfícies de visualização bidimensionais, temos de recorrer às projeções. Uma projeção representa sempre uma perda de informação. As projeções usadas são projeções planares geométricas, porque se projeta sobre um plano usando retas como projetantes. Estas projeções podem ser de dois tipos, paralelas ou perspectivas. Nas imagens que seguem mostram-se os dois exemplos de projeções aplicadas no contexto do nosso mundo 3D:

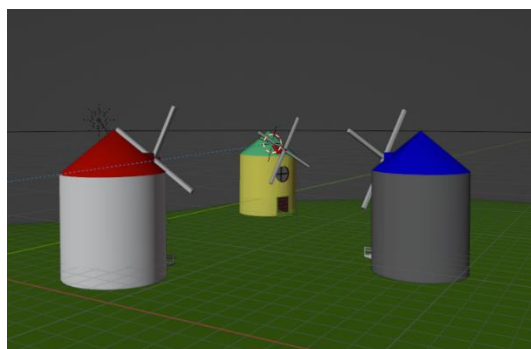


Figura 1: Projeção Perspetiva

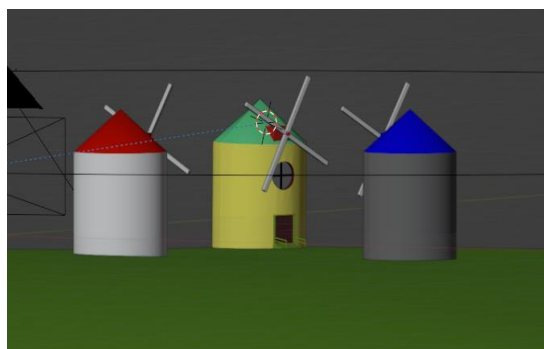


Figura 2: Projeção Paralela

A Figura 1 representa uma projeção perspetiva. Tal facto é facilmente visível através de inúmeros aspetos: as projetantes convergem no centro de projeção (encurtamento perspetivo), o volume de visualização é uma porção de um sólido, habitualmente um cone ou uma pirâmide, como neste caso, o centro de projeção não está contido no plano de projeção, e, neste tipo de projeção são mantidos os paralelismos das retas paralelas ao plano de projeção. Por fim, de ressaltar a existência de pontos de fuga na imagem, na medida em que as projeções das retas paralelas a uma direção, não paralela ao plano de projeção, convergem para o tal ponto de fuga.

A Figura 2 representa uma projeção paralela. Tal facto é facilmente visível através de inúmeros aspetos: as projetantes são paralelas entre si, o volume de visualização, tal

como se pode analisar, apresenta o formato de um paralelepípedo, o plano de projeção não é paralelo à direção de projeção e, neste tipo de projeção, são mantidos todos os paralelismos. Ao contrário da projeção anteriormente descrita, não existem os chamados pontos de fuga.

PONTO II:

Inicialmente, tendo em conta as definições “Default” do Blender com clip start a 0,1 e clip end a 1000, diminuámos o tamanho de end para 100:

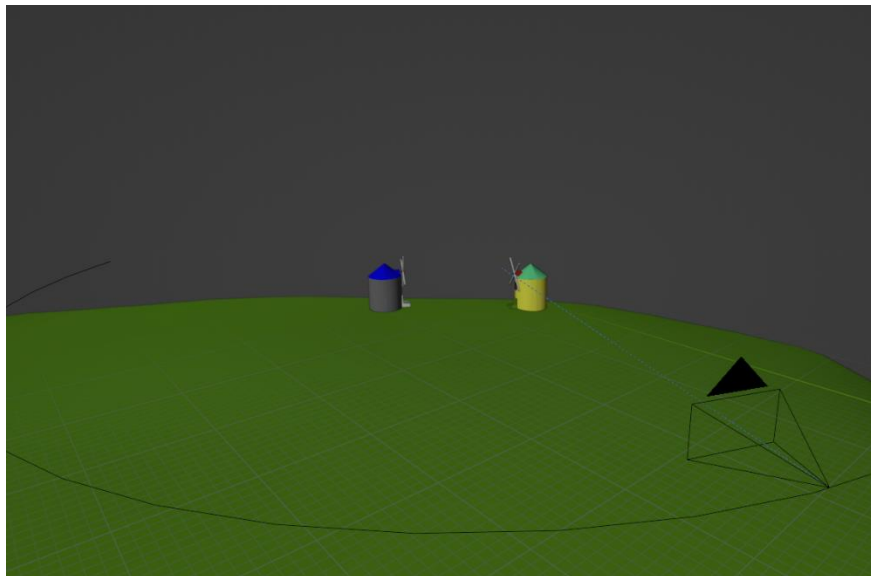


Figura 3: clip start-0,1 end-100

Com base na imagem original do ponto i), é possível verificar que parte do terreno, bem como do moinho vermelho, foram “cortados”, já que se encontram fora do alcance de visão. Tal acontece uma vez que, ao diminuirmos o tamanho de end no Blender, estamos a variar a posição do “**back clipping plane**” aproximando-o do “**front clipping plane**”. Como tal, quaisquer objetos que estejam para além do “back clipping plane” e, por sua vez, fora do volume de visualização (tendo em conta o centro de projeção), não serão visíveis.

Posteriormente, aumentámos o tamanho de clip start para 100:

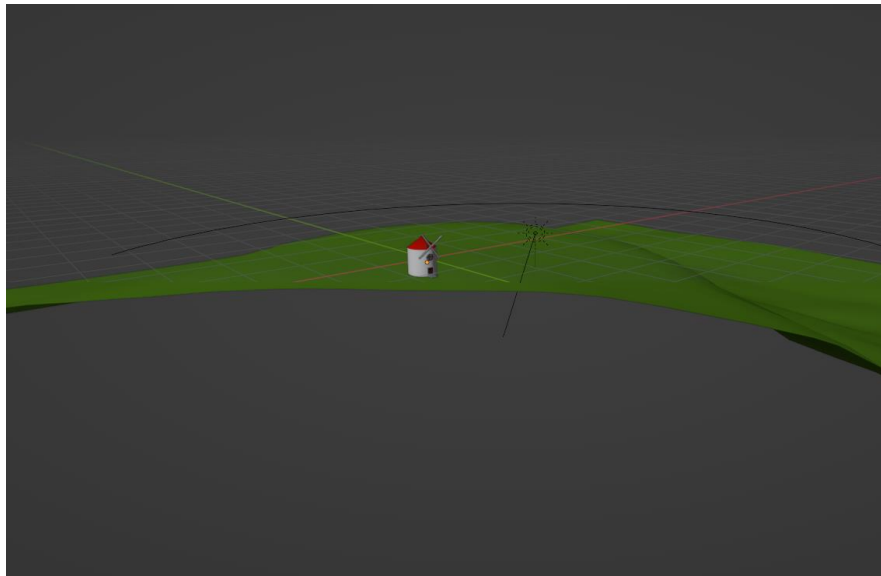


Figura 4: clip start-100 end-1000

Comparando esta figura com a anterior, os dois moinhos presentes na figura 1, bem como parte do terreno, foram “cortados”, estando visível apenas o moinho vermelho. Contudo, aparentam ter sido cortados pela frente.

Ao contrário da imagem anterior, estamos a aumentar o clip start resultando num avanço do “**front clipping plane**”, aproximando-o do “**back clippin plane**” que, por sua vez, novamente, tendo em conta o centro de projeção, leva a que só consigamos observar quaisquer objetos para lá do “front clipping plane” e até ao “back clippin plane” (tudo dentro do volume de visualização).

É de notar que em ambas as imagens a nossa “visão” é diminuída, uma vez que, ao aproximar o “**front clipping plane**” e o “**back clippin plane**” um do outro, estamos a diminuir o volume de visualização do objeto que representa a nossa visualização deste mundo 3D.

PONTO III:

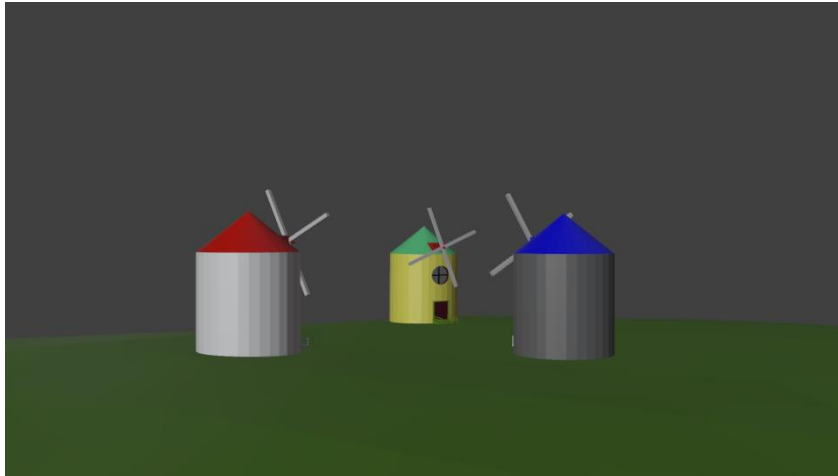


Figura 5 - Flat Shading - Workbench Render Engine - Todos os objetos com opção Flat Shading

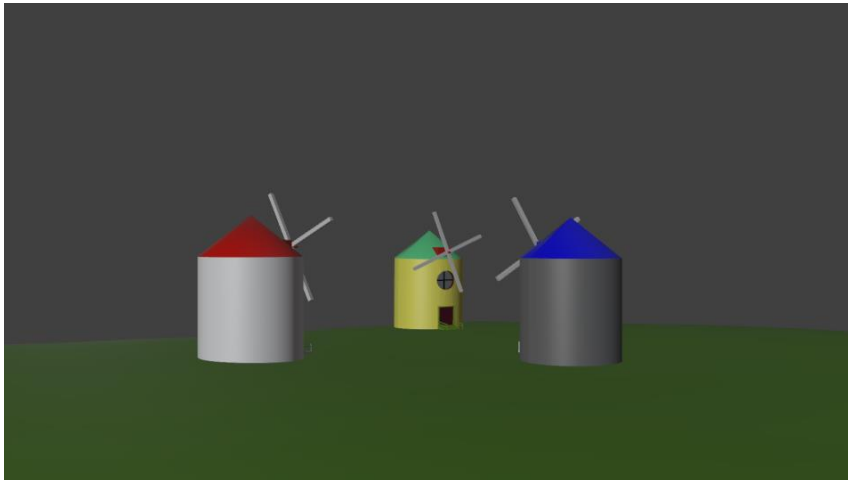


Figura 6 - Gouraud Shading - Workbench Render Engine - Todos os objetos com opção Auto Smooth Shading

A diferença existente entre as imagens deve-se aos dois métodos diferentes de Shading utilizados.

Com o algoritmo **Flat Shading** a cor de um polígono é sempre igual, dependendo do ângulo da normal deste. Quando os ângulos das normais entre polígonos adjacentes têm uma diferença suficiente, é perceptível onde um polígono acaba e o outro começa.

Já com o algoritmo **Gouraud Shading**, a intensidade da cor de cada pixel de um polígono é interpolada através das intensidades dos vértices deste, que são obtidas a partir da média das normais dos polígonos que partilham o vértice. Isto deixa cada pixel de um polígono ter uma cor diferente, o que causa um Shading mais suave.

“QUEM FEZ O QUE?”:

- A Parte 1 e 2 foram realizadas pelos 3 elementos do grupo de forma igual;
- A Parte 3 relativa ao relatório foi dividida por pontos, onde cada elemento realizou um deles:

-> Ponto i) – Tomás Piteira;

-> Ponto ii) – Henrique Catarino;

-> Ponto iii) – Miguel Nunes;

LINKS UTILIZADOS:

- <https://acervolima.com/diferenca-entre-projecao-paralela-e-em-perspectiva-em-computacao-grafica/>
- <https://docs.blender.org/manual/en/latest/index.html>
- https://graphics.fandom.com/wiki/Gouraud_shading
- https://www.youtube.com/watch?v=d_pWhjH7sU&t=75s

- FIM -