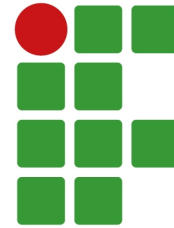


OpenGL:

Teste de Oclusão (*z-buffer*)



**INSTITUTO
FEDERAL**

São Paulo

Câmpus
Presidente Epitácio

Prof. Dr. Bruno César Vani

Notas de Aula de Computação Gráfica

BCC / 2020

Conteúdo e Objetivos

- Compreender o uso da profundidade (coordenada z), inicialmente no caso da projeção ortogonal;
- Compreender o algoritmo de teste de profundidade (*depth test* ou z -buffer) e sua utilidade;
- Utilizar configurações de *depth test* no OpenGL;
- Resolver exercício de fixação.

Introdução / Contexto

- Em aulas anteriores, trabalhamos com projeção ortogonal, desprezando-se os valores de z na maioria dos casos;
- Começamos compreender o uso da coordenada z para o caso específico do *clipping*;
- Nesta aula, vamos utilizar a coordenada z como forma de representar a profundidade em cenas, inicialmente, no caso da projeção ortogonal.

Ideia do algoritmo z-buffer:

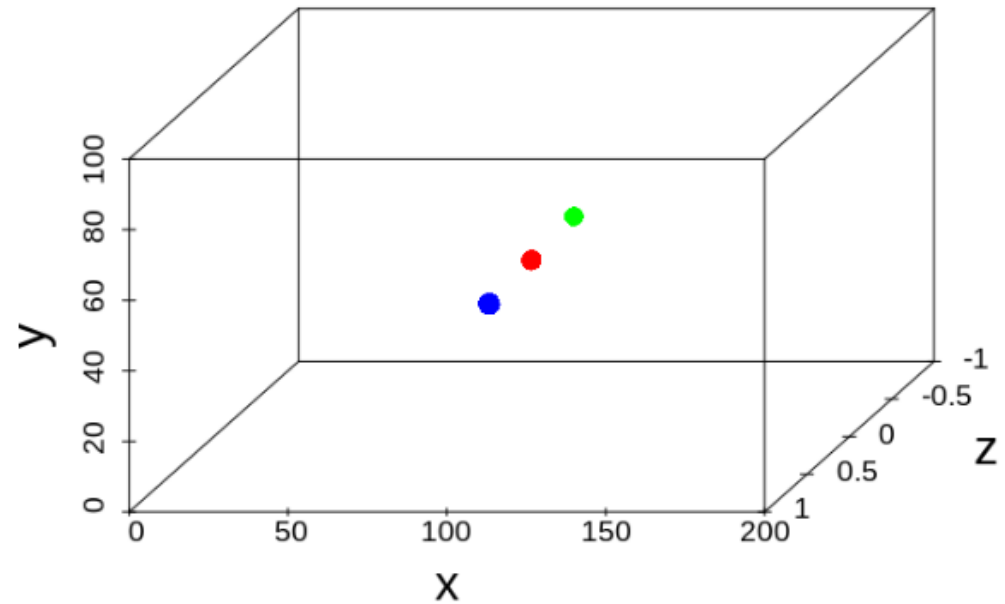
- Considere o seguinte problema:
 - Imagine três objetos em uma cena, cada um de uma cor;
 - Imagine que eles estão dispostos com sobreposição, ou seja, “um na frente do outro”;
- Qual objeto se sobrepõe ao outro? Em que ordem eles aparecerão primeiro na tela? É mesmo necessário desenhar todos os objetos?
 - Em uma cena complexa, pode-se recorrer à coordenada z para atribuir a profundidade;
 - O algoritmo z-buffer tratará de não desenhar as superfícies de objetos escondidos por outros objetos (oclusos).

Outros nomes para o z-buffer

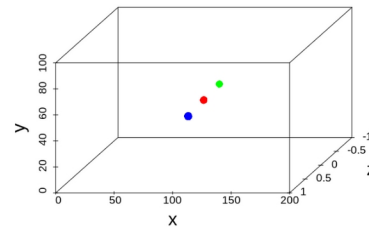
- Outros nomes aparecem na literatura:
 - Teste de profundidade (*depth test*);
 - *Z-index*;
 - Remoção de superfícies escondidas (*hidden surface removal*);
 - Teste (ou determinação) de visibilidade;
 - Etc.

Exemplo em um *viewing box*

- Para fins de testes e aprendizagem, utilizando o *viewing box* definido conforme a figura:
 - Considere 3 pontos com os mesmos valores x,y, variando apenas no valor de z
- Os pontos mais próximos da *viewing face* terão o valor z maior



Algoritmo z-buffer



- O algoritmo z-buffer é constituído por um bloco de memória na GPU contendo os valores de z para cada pixel:
- Algoritmo:
 - Quando um pixel é atribuído, é verificado se o valor z deste pixel é superior ao atual:
 - Caso sim, o pixel recebe o componente de cor RGB correspondente ao novo valor z;
 - Caso contrário, trata-se de um valor ocluído, logo, mantém-se o componente de cor RGB correspondente ao valor z existente.

Experimento no OpenGL

- No OpenGL, é necessário habilitar o *depth test*;
 - `glEnable(GL_DEPTH_TEST);`
- De maneira similar ao buffer de cor de preenchimento, limpa-se o a memória do *depth test* com:
 - `glClear(GL_DEPTH_BUFFER_BIT);`
- Código-fonte: DepthTest

Exercício

1) Tente reproduzir o objeto a seguir. Dica de solução: modifique a função desenhar círculo 2D para que considere também a coordenada z e utilize o *depth test* para desenhar círculos menores mais a frente.

