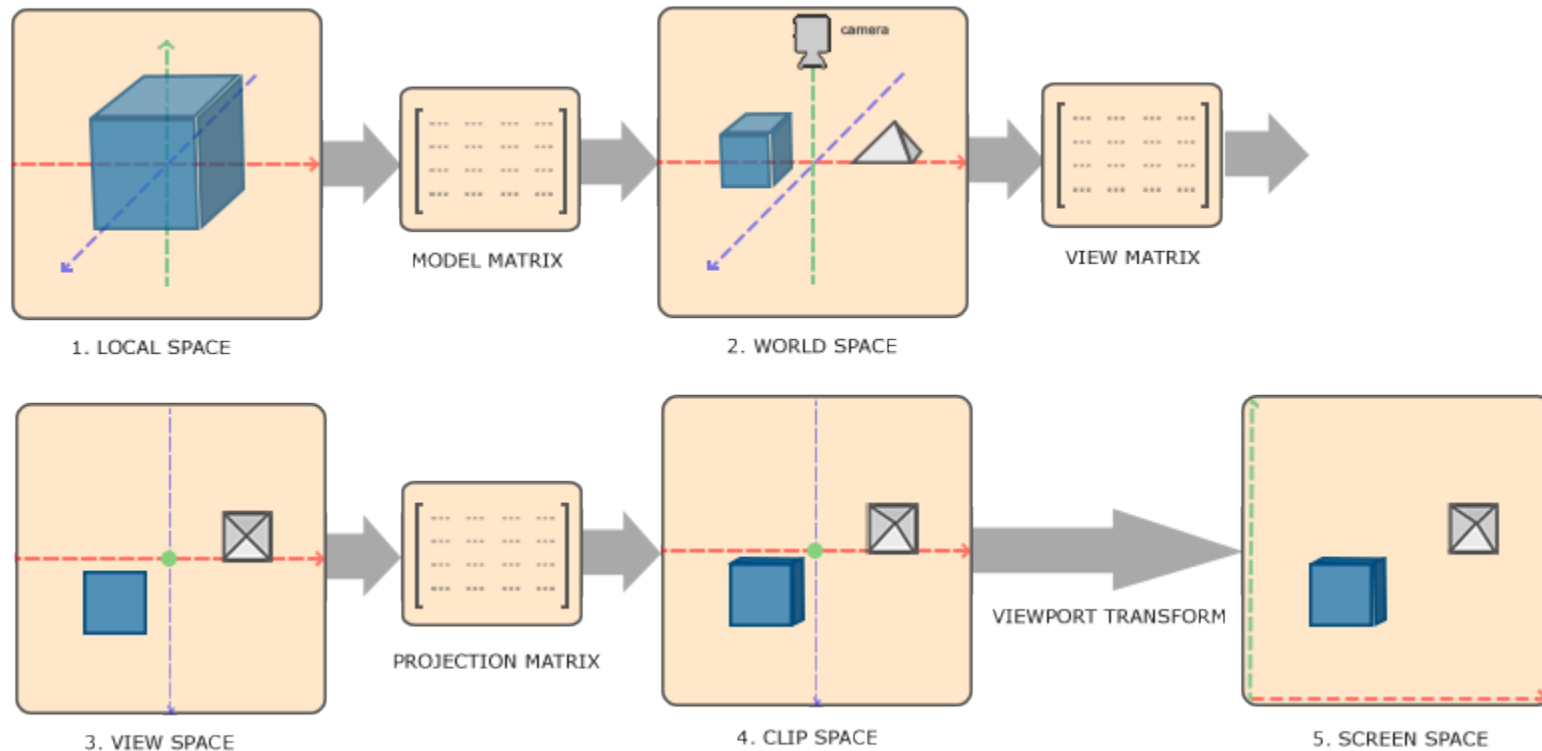


Sistema de Coordenadas Projeção Ortográfica

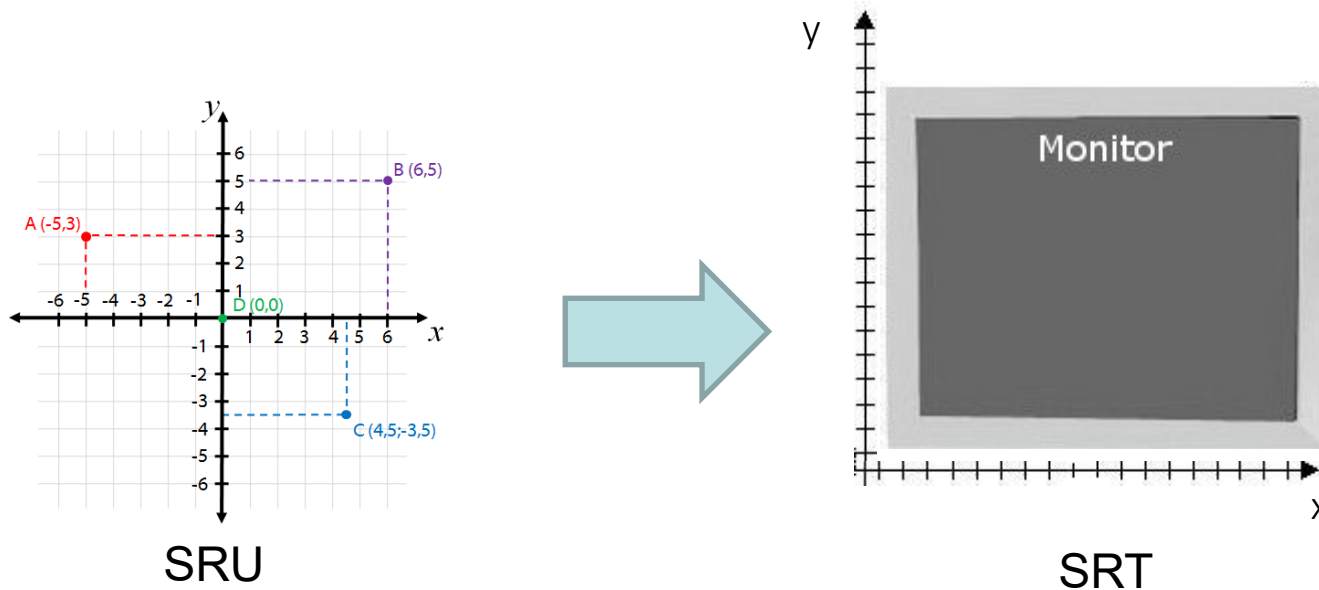
por Rossana B Queiroz

Sistemas de Coordenadas



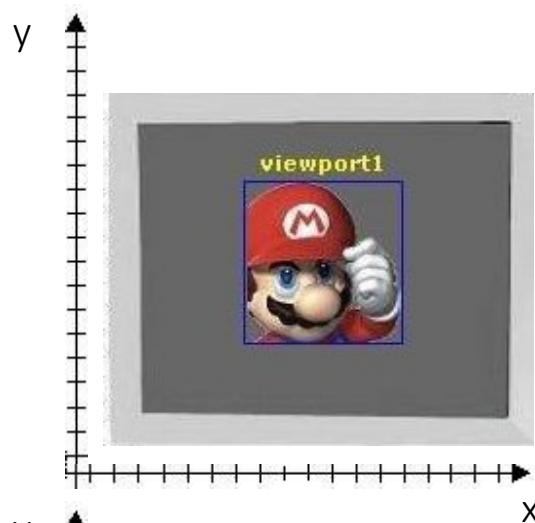
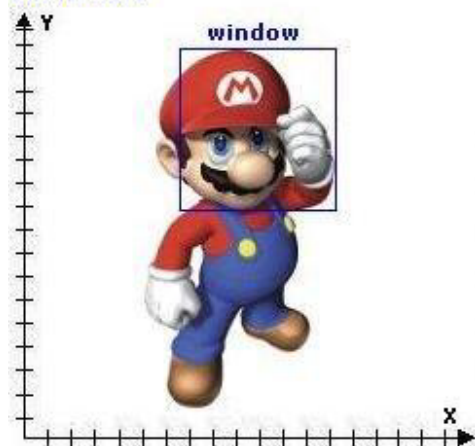
Sistemas de Coordenadas

- Coordenadas locais (*local space*)
- Coordenadas de Universo (*world space*) – Sistema de Referência do Universo (SRU)
- Coordenadas de camera (*view space*)
- Espaço de Recorte (*clip space*)
- Coordenadas de Tela (*screen space*) – Sistema de Referência de Tela (SRT)

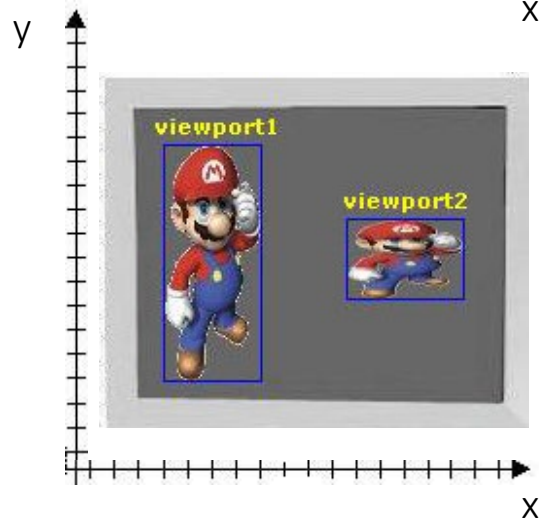
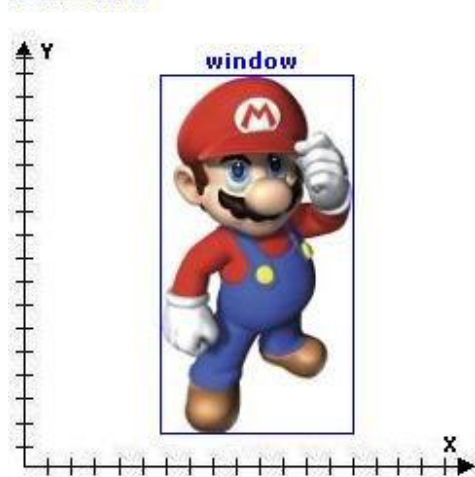


SRU vs. SRT

EXEMPLO 1

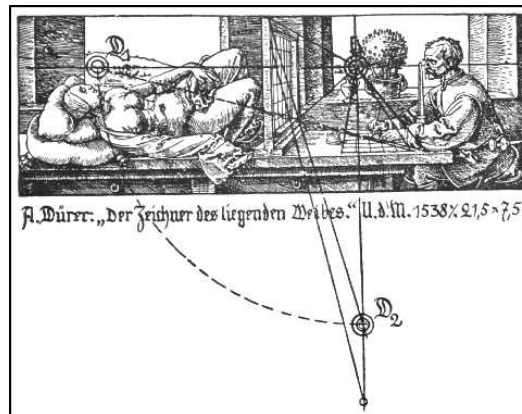


EXEMPLO 2



Perspectivas

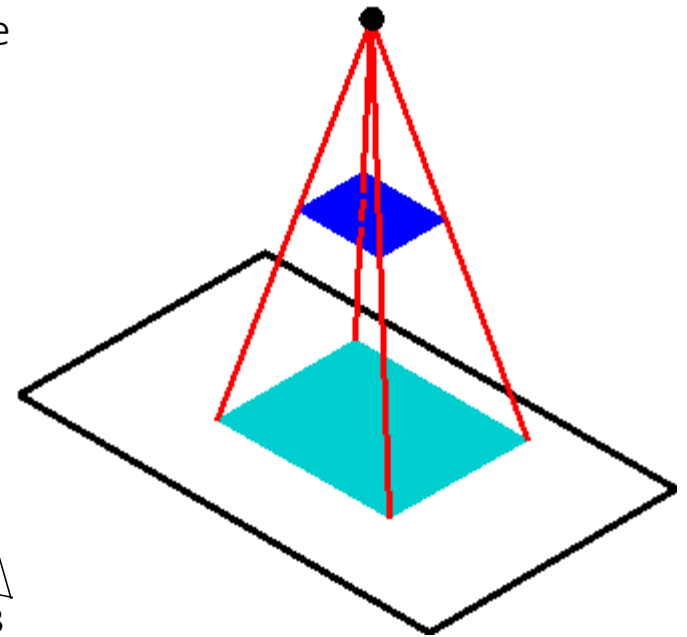
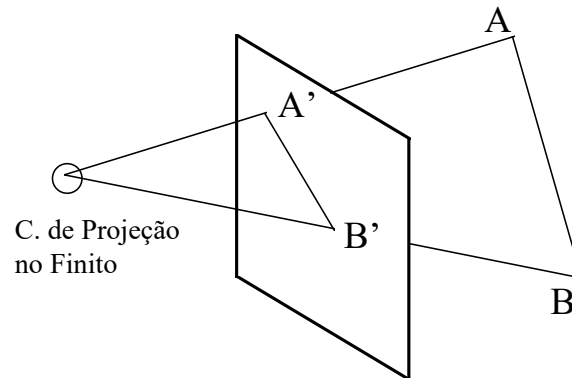
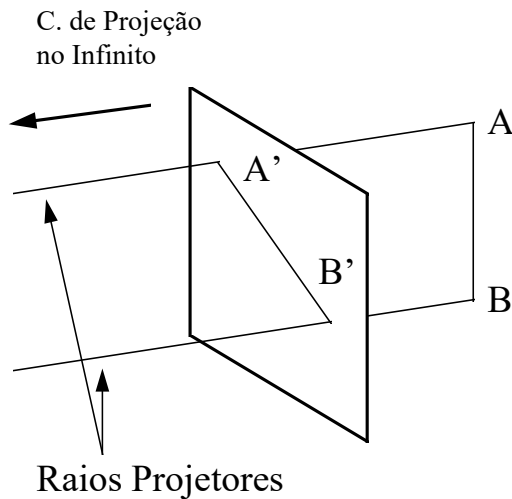
- A palavra perspectiva vem do latim - *Perspicere* (ver através de)
- Se você se colocar atrás de uma janela envidraçada e, sem se mover do lugar, riscar no vidro o que está "vendo através da janela", terá feito uma perspectiva
- Perspectiva é a representação gráfica que mostra os objetos como eles aparecem a nossa vista, com três dimensões.



Projeções Paralelas e Perspectivas

As projeções planares paralelas e perspectivas diferem com relação a distância do plano de projeção ao centro de projeção:

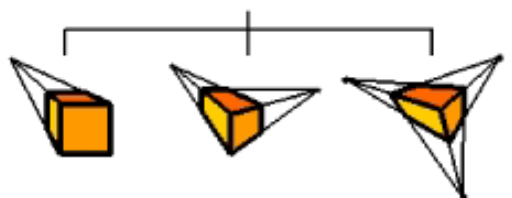
- se a distância é finita, a projeção é perspectiva, e
- se a distância é infinita, a projeção é paralela



Projeções Paralelas e Perspectivas

Perspectivas

PROJEÇÕES CÔNICAS



Um ponto
de fuga

Dois pontos
de fuga

Três pontos
de fuga

Paralelas

PROJEÇÕES CILÍNDRICAS



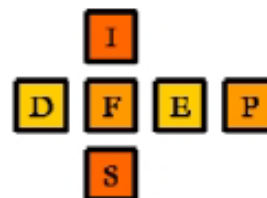
Oblíqua

Ortográfica



Cavalera

Axonométrica



Múltiplas vistas ortográficas



Isométrica

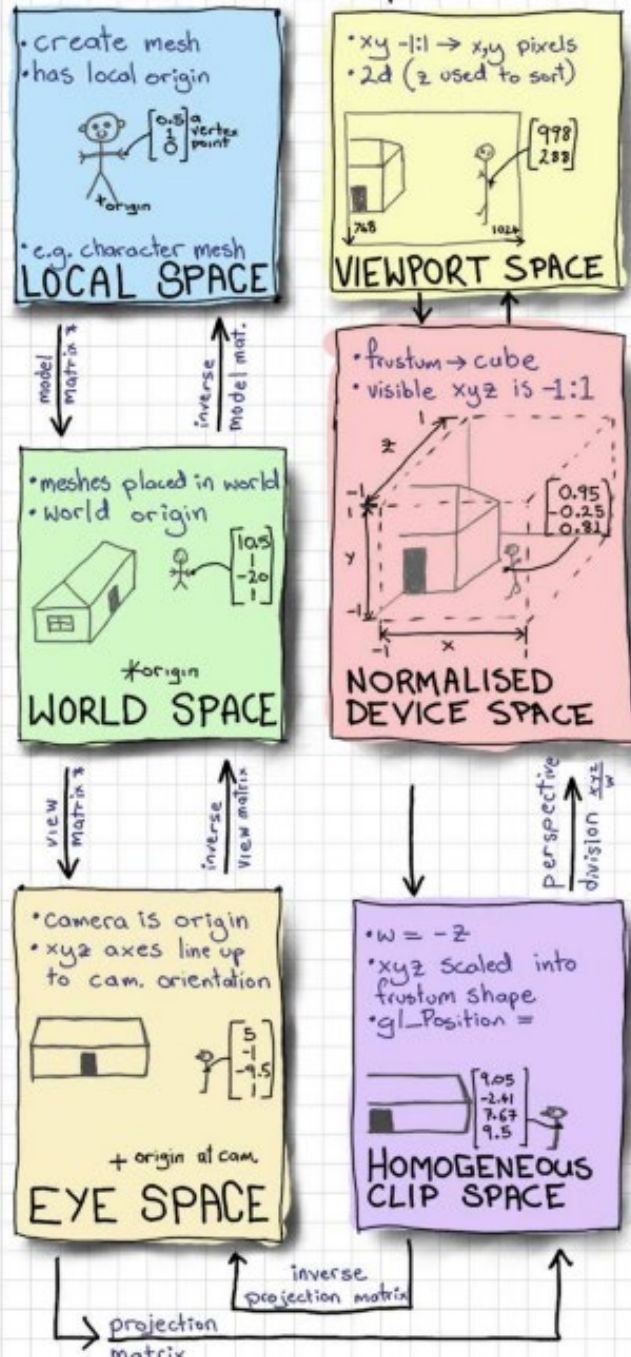


Dimétrica



Trimétrica

3d Transformation Pipeline



Como funciona

Manipulado no **vertex shader** usando 3 matrizes:

1. **Model Matrix:** posiciona e orienta os vértices os vértices de um objeto na cena
2. **View Matrix:** posiciona e orienta os vértices em relação à câmera
3. **Perspective Matrix:** adiciona perspectiva; sensação de profundidade

`glm::mat4 model;`

`glm::mat4 view;`

`glm::mat4 projection;`

GLM

- Biblioteca auxiliar para trabalhar com matemática (*header-only*)

- <https://glm.g-truc.net/0.9.8/index.html>

```
#include <glm/glm.hpp>
#include <glm/gtc/matrix_transform.hpp>
#include <glm/gtc/type_ptr.hpp>
```

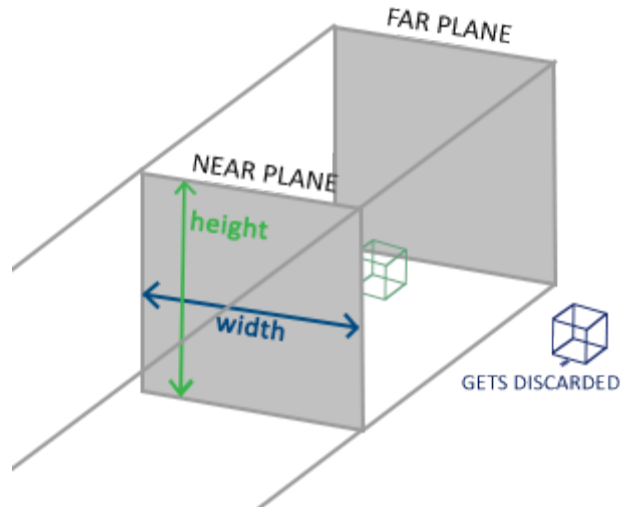
Matriz de View

- Define a posição e orientação do observador (câmera)
- Transformações simples ou *LookAt*

```
view = glm::translate(view, glm::vec3(0.0f, 0.0f, -1.0f));
```

Não é necessário definir o view utilizando a câmera ortográfica.

Projeção Ortográfica



```
ortho = glm::ortho(-10.0f, 10.0f, -10.0f, 10.0f, -1.0f, 1.0f);
```

Passando a matriz para o shader

- Variável do tipo **uniform**, portanto:

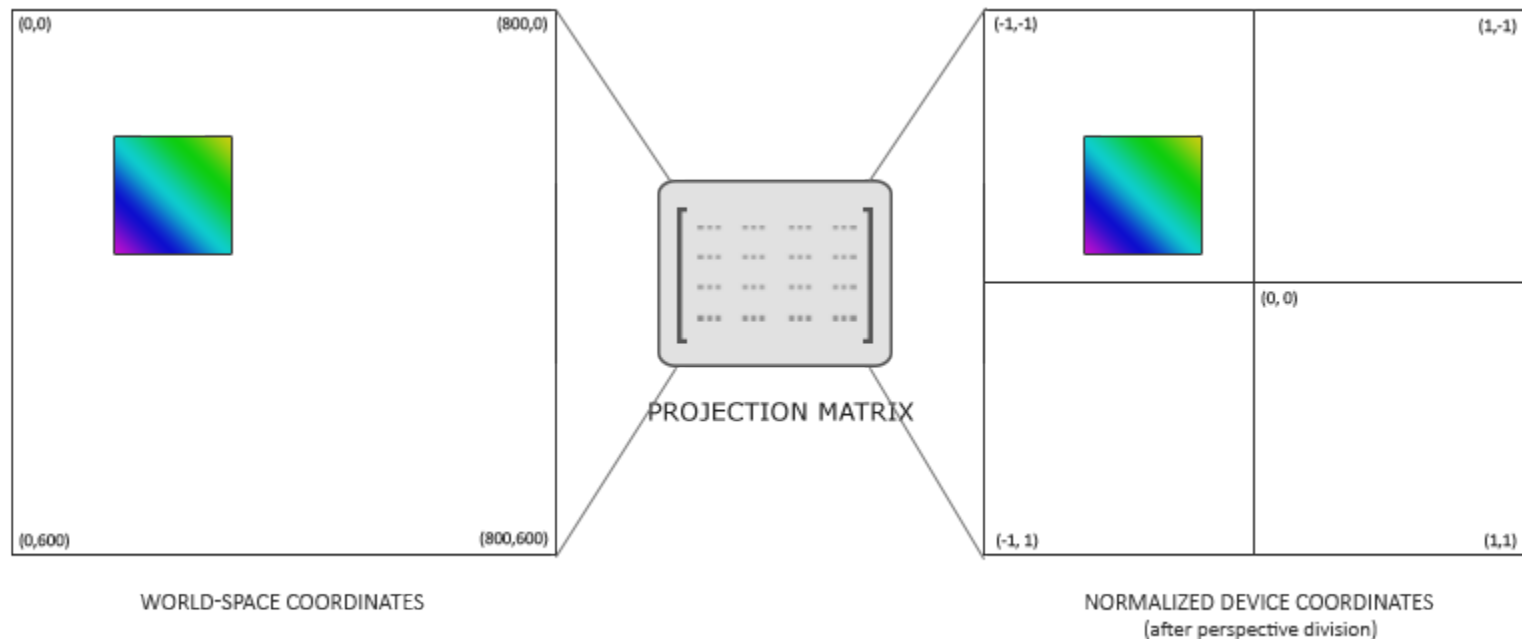
```
glm::mat4 projection = glm::mat4(1);  
projection = glm::ortho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);  
  
GLint projLoc = glGetUniformLocation(shader.ID, "projection");  
glUniformMatrix4fv(projLoc, 1, FALSE, glm::value_ptr(projection));
```

- No vertex shader:

```
...  
uniform mat4 projection;  
...  
void main()  
{  
    gl_Position = projection * vec4(position, 1.0);  
}
```

Projeção Ortográfica

```
glm::mat4 projection = glm::ortho(0.0f, 800.0f, 600.0f, 0.0f, -1.0f, 1.0f);
```



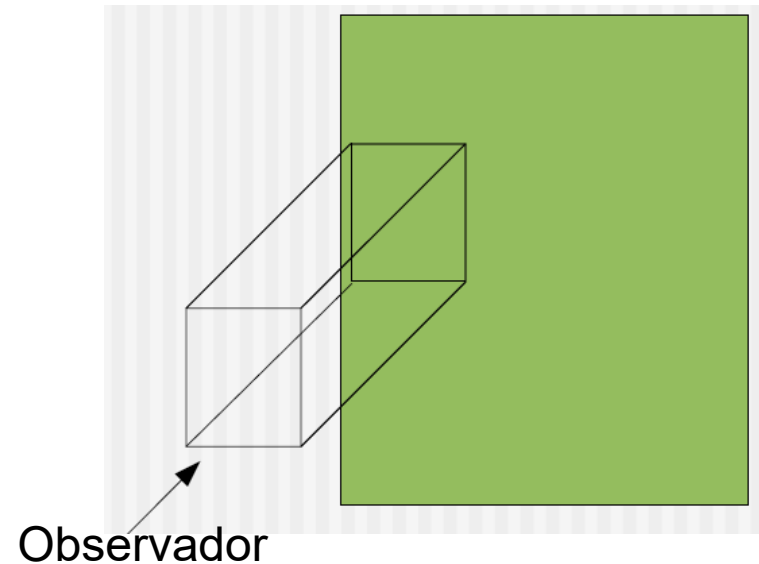
Scrolling

- Quando as dimensões do mundo são maiores que a da câmera
- A câmera (*window – ortho*) se movimenta com o personagem
- Isso acontece quando a *viewport* não tem tamanho adequado para mostrar todo o cenário



Scrolling

- Conceito de uma **câmera sintética**
- Em 2D, trata-se de uma **câmera ortográfica**, ou seja, sem perspectiva



- Scrolling em OpenGL??
 - Em que comando devemos mexer??

Referências bibliográficas

- Cohen, M., & Manssour, I. H. (2006). *OpenGL: uma abordagem prática e objetiva*. Novatec editora.
- Slides sobre CG dos professores: Christian Hofsetz, Cristiano Franco, Marcelo Walter, Soraia Musse, Leandro Tonietto e Rafael Hocevar.
- Leituras obrigatórias:
 - <https://learnopengl.com/#!Getting-started/Transformations>
 - <https://learnopengl.com/#!Getting-started/Coordinate-Systems>