

Aula – Computação Gráfica

OpenGL 2D

Slides para uso pessoal e exclusivo durante o período de aula. Distribuição ou qualquer uso fora do escopo da disciplina é expressamente proibido.

1

1

OpenGL API - Fixed-Function vs. Programável

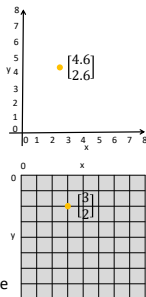
- Fixed-Function API é mais simples de prototipar
 - Implementa a maioria da álgebra linear para mover objetos
- Programmable API é mais moderna
 - Usa *shaders* por trás dos panos
 - Provê as funções da Fixed-Function API por compatibilidade

2

2

Sistema de Coordenadas

- Cartesiano na matemática e na engenharia
 - Tipicamente modelado com ponto flutuante
 - Tipicamente com
 - X crescendo para a direita
 - Y crescendo para cima
- Coordenadas do Monitor (Física)
 - Utiliza inteiros
 - Tipicamente com
 - X crescendo para a direita
 - Y crescendo para baixo
 - 1 unidade = 1 pixel
- Para evitar pensar como nos monitores
 - Escolha/defina seu sistema convenientemente

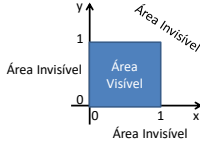


3

3

Sistema de Coordenadas

- Escolha uma convenção para seu sistema
 - Para ficar intuitivo
 - X crescendo para a direita
 - Y crescendo para cima
 - Defina a área visível
 - Unidades são baseadas no tamanho da janela
 - Área visível será esticada para preencher a janela
 - Unidades são, portanto, porcentagens do tamanho da janela



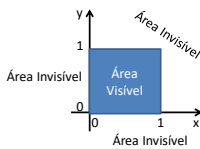
4

4

Sistema de Coordenadas

- Defina seu sistema de coordenadas com a matriz de projeção

```
glMatrixMode(GL_PROJECTION); // Seleccione a matriz de projecao
glOrtho( 0, // coordenada X da aresta esquerda
         1, // coordenada X da aresta direita
         0, // coordenada Y da aresta de baixo
         1, // coordenada Y da aresta de cima
        -1, // coordenada Z do plano "near"
         1); // coordenada Z do plano "far"
```

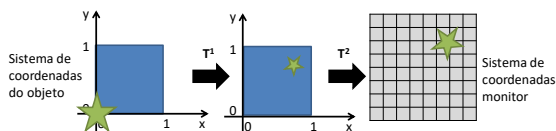


5

5

Sistema de Coordenadas (SC)

- Existem duas forma de pensar
 - Construir o mundo virtual no SC global
 - Definido no slide anterior
 - Inconveniente, pois objeto iguais poderão ter vértices diferentes
 - Construir cada objeto do mundo em seu próprio SC
 - Definir um SC (implícito) para cada objeto
 - Transformar (T) cada objeto para sua forma e pose no SC global



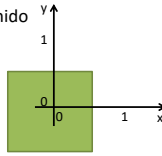
6

6

Funções Básicas de OpenGL

- Como desenhar um quadrado no SC definido

```
glBegin(GL_QUADS);
    glVertex2f(-.5, -.5);
    glVertex2f( .5, -.5);
    glVertex2f( .5,  .5);
    glVertex2f(-.5,  .5);
glEnd();
```



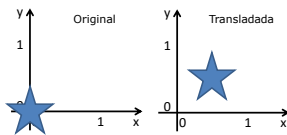
- A ordem dos vértices é importante
 - Vértices devem ser definidos no sentido anti-horário em relação ao visualizador

- Isso determina a direção da normal
- Normal na direção errada impossibilita a visualização
- Sentido anti-horário é consistente com a regra da mão direita

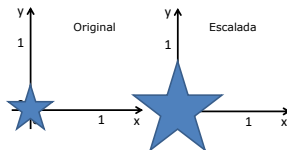


7

Transformações



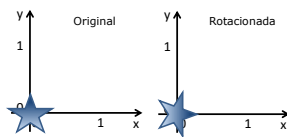
```
glTranslatef(.5, .5, 0);
```



```
glScalef(2, 2, 1);
```

8

Transformações



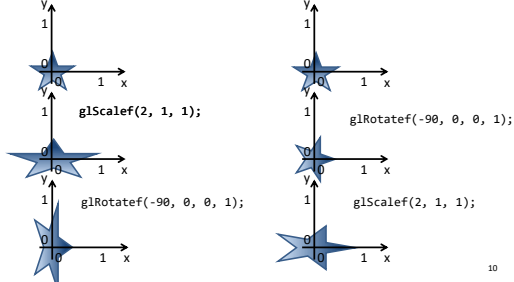
```
glRotatef(-90, 0, 0, 1);
```

Ângulos positivos rotacionariam no sentido anti-horário (regra da mão direita)

9

Transformações

- Transformações podem ser compostas
- A ordem deve ser respeitada (não comutativa)



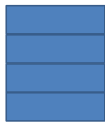
10

10

Transformações

- Para mapear do SC do objeto para o SC global
 - Utilize a matriz de modelagem (modelview matrix)

```
glMatrixMode(GL_MODELVIEW);
// As chamadas de transformacoes (glTranslatef,
glRotatef, etc) seguem essa chamada da modelview
```



GL_MODELVIEW



GL_PROJECTION

11

11

Transformações

- Para mapear do SC do objeto para o SC global

```
glLoadIdentity(); // I
glTranslatef(x, y, 0);
glRotatef(theta1, 0,0,1);
glBegin(GL_POINTS);
    glVertex3f(x, y, 0);
glEnd();

glLoadIdentity(); // I
glRotatef(theta1, 0,0,1);
glTranslatef(x, y, 0);
glBegin(GL_POINTS);
    glVertex3f(x, y, 0);
glEnd();
```



GL_MODELVIEW



GL_MODELVIEW

12

12

Transformações

- Para mapear do SC do objeto para o SC global

```
glLoadIdentity(); // I
glTranslatef(x, y, 0); // T1
glRotatef(theta1, 0,0,1);
glBegin(GL_POINTS);
    glVertex3f(x, y, 0);
glEnd();
```



```
glLoadIdentity(); // I
glRotatef(theta1, 0,0,1); // R1
glTranslatef(x, y, 0);
glBegin(GL_POINTS);
    glVertex3f(x, y, 0);
glEnd();
```



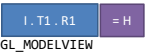
13

13

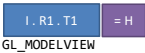
Transformações

- Para mapear do SC do objeto para o SC global

```
glLoadIdentity(); // I
glTranslatef(x, y, 0); // T1
glRotatef(theta1, 0,0,1); // R1
glBegin(GL_POINTS);
    glVertex3f(x, y, 0);
glEnd();
```



```
glLoadIdentity(); // I
glRotatef(theta1, 0,0,1); // R1
glTranslatef(x, y, 0); // T1
glBegin(GL_POINTS);
    glVertex3f(x, y, 0);
glEnd();
```



14

14

Transformações

- Para mapear do SC do objeto para o SC global

```
glLoadIdentity(); // I
glTranslatef(x, y, 0); // T1
glRotatef(theta1, 0,0,1); // R1
glBegin(GL_POINTS);
    glVertex3f(x, y, 0); // v
glEnd();
```



```
glLoadIdentity(); // I
glRotatef(theta1, 0,0,1); // R1
glTranslatef(x, y, 0); // T1
glBegin(GL_POINTS);
    glVertex3f(x, y, 0); // v
glEnd();
```



15

15

Transformações

- Navegue nos SCs com a pilha de matrizes do OpenGL
 - Permite desenhar um objeto com uma transformação
 - E depois retornar a transformação anterior
 - Push e Pop

```
glPushMatrix(); // Salva o estado atual da pilha
// Faz as transformações necessárias para desenhar um objeto
(glTranslatef, etc)
// Desenha o objeto (glBegin, glVertex, etc)
glPopMatrix(); // Carrega o ultimo estado salvo

// Continua a modelagem dos outros objetos considerando o último SC
```

16

16

Transformações

- Para mapear do SC do objeto para o SC global

```
glLoadIdentity(); // I
glTranslatef(x, y, 0); // T1
glPushMatrix(); // H1
glRotatef(theta1, 0,0,1); // R1
glBegin(GL_POINTS);
glVertex3f(x1, y1, 0); // v1
glEnd();
glPopMatrix();
glRotatef(theta2, 0,0,1); // R2
glBegin(GL_POINTS);
glVertex3f(x2, y2, 0); // v2
glEnd();
```

Diagram illustrating the transformation process:

Initial state: $I \cdot T1 \cdot R1 = H$ (GL_MODELVIEW)

After first transformation: $v1' = H \cdot v1$

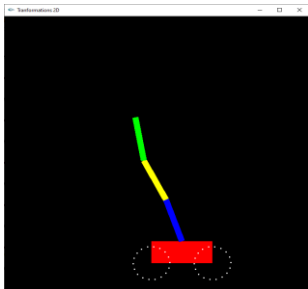
After second transformation: $v2' = H \cdot v2$

17

17

Exemplo de Composição de Transformações

- Suponha que se queira desenhar o robô abaixo



18

18

Exemplo de Composição de Transformações

- Criar uma função de desenho de um retângulo (DesRect)
 - Escolher o origem do sistema de coordenadas do objeto
 - Escolhido o centro da base
 - Com isso, o objeto deve ser desenhado para que seja assim

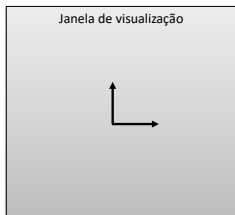


19

19

Exemplo de Composição de Transformações

- Criar uma função de desenho de um retângulo (DesRect)
 - Escolher o origem do sistema de coordenadas do objeto
 - Escolhido o centro da base
 - Com isso, o objeto deve ser desenhado para que seja assim

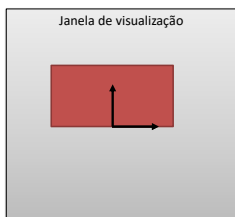


20

20

Exemplo de Composição de Transformações

- Criar uma função de desenho de um retângulo (DesRect)
 - Escolher o origem do sistema de coordenadas do objeto
 - Escolhido o centro da base
 - Com isso, o objeto deve ser desenhado para que seja assim

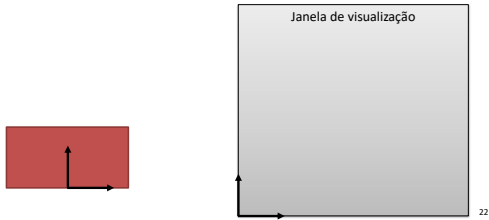


21

21

Exemplo de Composição de Transformações

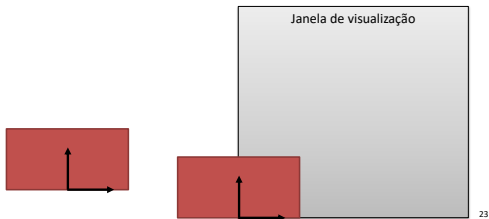
- Criar uma função de desenho de um retângulo (DesRect)
 - Escolher o origem do sistema de coordenadas do objeto
 - Escolhido o centro da base
 - Com isso, o objeto deve ser desenhado para que seja assim



22

Exemplo de Composição de Transformações

- Criar uma função de desenho de um retângulo (DesRect)
 - Escolher o origem do sistema de coordenadas do objeto
 - Escolhido o centro da base
 - Com isso, o objeto deve ser desenhado para que seja assim

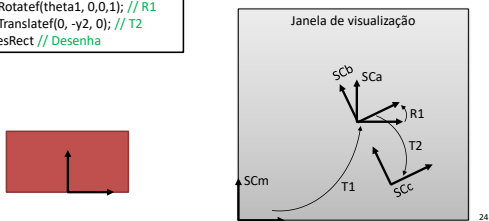


23

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas

```
glLoadIdentity(); // SCm
glTranslatef(x1, y1, 0); // T1
glRotatef(theta1, 0,0,1); // R1
glTranslatef(0, -y2, 0); // T2
DesRect // Desenha
```

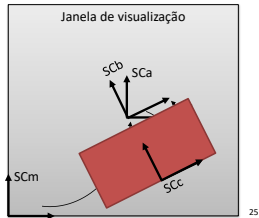


24

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas

```
glLoadIdentity(); // SCm
glTranslatef(x1, y1, 0); // T1
glRotatef(theta1, 0,0,1); // R1
glTranslatef(0, -y2, 0); // T2
DesRect // Desenha
```



25

25

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas



26

26

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas



27

27

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas

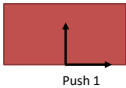


28

28

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas
 - Armazene o sistema de coordenadas mais conveniente (push)

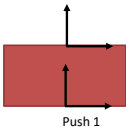


29

29

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas
 - Armazene o sistema de coordenadas mais conveniente (push)

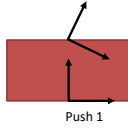


30

30

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas
 - Armazene o sistema de coordenadas mais conveniente (push)

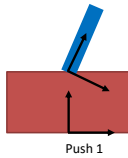


31

31

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas
 - Armazene o sistema de coordenadas mais conveniente (push)

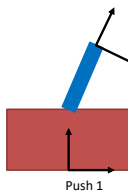


32

32

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas
 - Armazene o sistema de coordenadas mais conveniente (push)

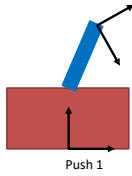


33

33

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas
 - Armazene o sistema de coordenadas mais conveniente (push)

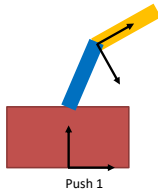


34

34

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas
 - Armazene o sistema de coordenadas mais conveniente (push)

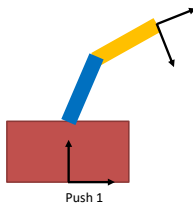


35

35

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas
 - Armazene o sistema de coordenadas mais conveniente (push)

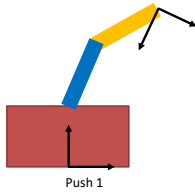


36

36

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas
 - Armazene o sistema de coordenadas mais conveniente (push)

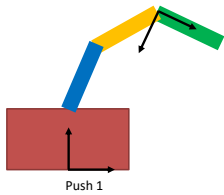


37

37

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas
 - Armazene o sistema de coordenadas mais conveniente (push)

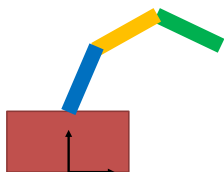


38

38

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas
 - Retorne ao sistema salvo (pop)

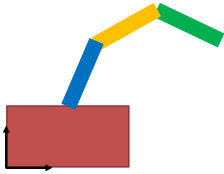


39

39

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas
 - Retorne ao sistema salvo (pop)

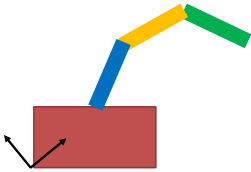


40

40

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas
 - Retorne ao sistema salvo (pop)

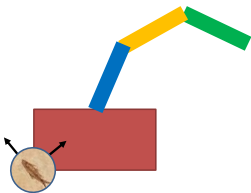


41

41

Exemplo de Composição de Transformações

- Antes de chamar a função de desenho
 - Transforme o sistema de coordenadas
 - Retorne ao sistema salvo (pop)



42

42

Animação

- Mostrar rapidamente uma sequência de imagens
- Tirar várias fotos do seu mundo virtual
 - Mostrar na sequência, uma após a outra



[Kineograph](#)
https://commons.wikimedia.org/wiki/File:Linnet_kineograph_1886.jpg
Public Domain

43

43

Perguntas ?????

44

44