openGL
Robot

DEI – 2019/2020 - Computação Gráfica

Jorge Henriques
Evgheni Polisciuc

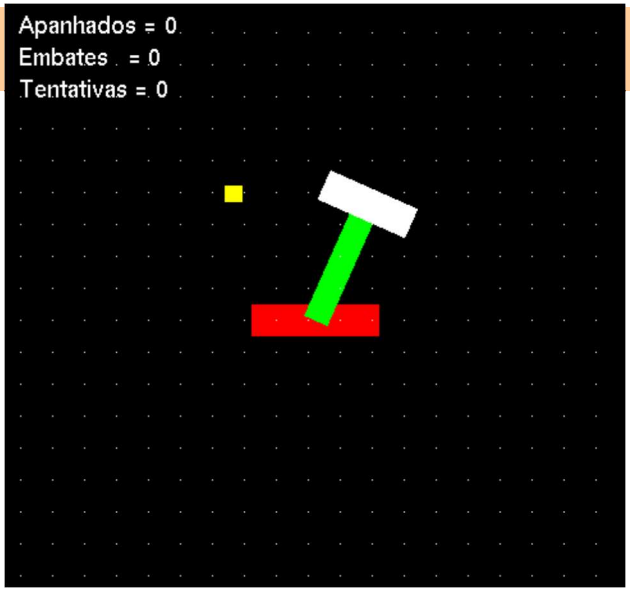
Objectivos

- Estudar como o OpenGL permite implementar operações básicas de transformação :
  - Translação
  - Rotação
  - Escala
  - Ordem das transformações
  - Combinação de transformações
  - Matriz model_View
  - push / pop

Objectivos

- Além disso
  - Definição e criação de janelas de visualização
  - Sistema de coordenadas real e sistema de coordenadas da janela de visualização
  - Bibliotecas típicas em OpenGl e sua utilização
  - Utilização de variáveis e comandos em OpenGL
  - Desenho de primitivas básicas
  - Especificação de cores
  - Gestão de eventos (teclado)
  - Desenho de caracteres (fontes bitmap)

Objectivos



JOGO – Regras

- ***Quadrado***= parte de uma posição semi-aleatória,

- ***Robot***
  - um **suporte**, que se move apenas na horizontal (setas esquerda e direita, por exemplo)
  - uma **ligação**, que roda em torno do suporte (utilizando por exemplo as setas cima e baixo)
  - uma **extremidade=escudo** que roda livremente

- O quadrado (amarelo) percorre o ecrã podendo acontecer uma de duas coisas:
  - O jogador apanha-o com a extremidade (**apanhados++**)
  - O quadrado embate na base (**embates++**)
- Final
  - O jogo termina quando ***embates>=apanhados+3 ??***
  - Ter um numero de créditos inicial que é decrementado/decrementado.
  - O jogo termina quando ??????

openGL

1. Desenhar um rectangulo = "os objectos"

- Objecto complexo constituído por 3 objectos simples
  - Quadrados

- Transformações = operações sobre os objectos
  - Translações, rotações, escalas

  - Atenção:
    - Operações podem não ser comutativas:
      Ex: (Rotação + Translação) ***DIFERENTE DE*** (Translação + Rotação) !!!
    - Em openGl a "***ordem é inversa***" (baixo para cima) !!!

    - Às vezes dá jeito que as transformações sejam ***globais*** a todos os objectos

openGL

2. Apanhar os objectos = *coordenadas dos objectos*

- Como saber se:
  - quadrado ***embate*** no suporte ?
  - quadrado ***apanhado*** pela extremidade ?

- Extremidade
  - Como saber "*aonde anda*" a extremidade ??

openGL

openGL

2. Coordenadas dos objectos

- **GL_MODELVIEW_MATRIX**
  - **Matriz que integra: "Modelos" + "Visualização"**

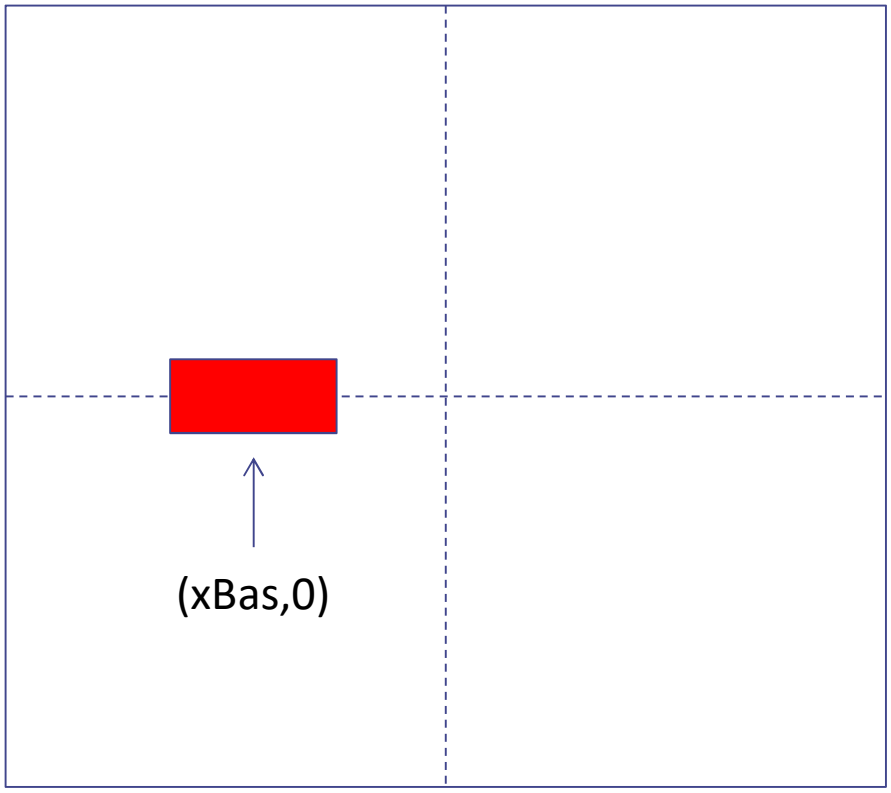
- GLfloat Matriz[4][4];
- glGetFloatv(GL_MODELVIEW_MATRIX, &Matriz[0][0]);

$$M = \begin{bmatrix} * & * & * & tx \\ * & * & * & ty \\ * & * & * & tz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Em openGL matriz modelView
- ***Transposta***
- *Normalizada* ***[-1,1]***

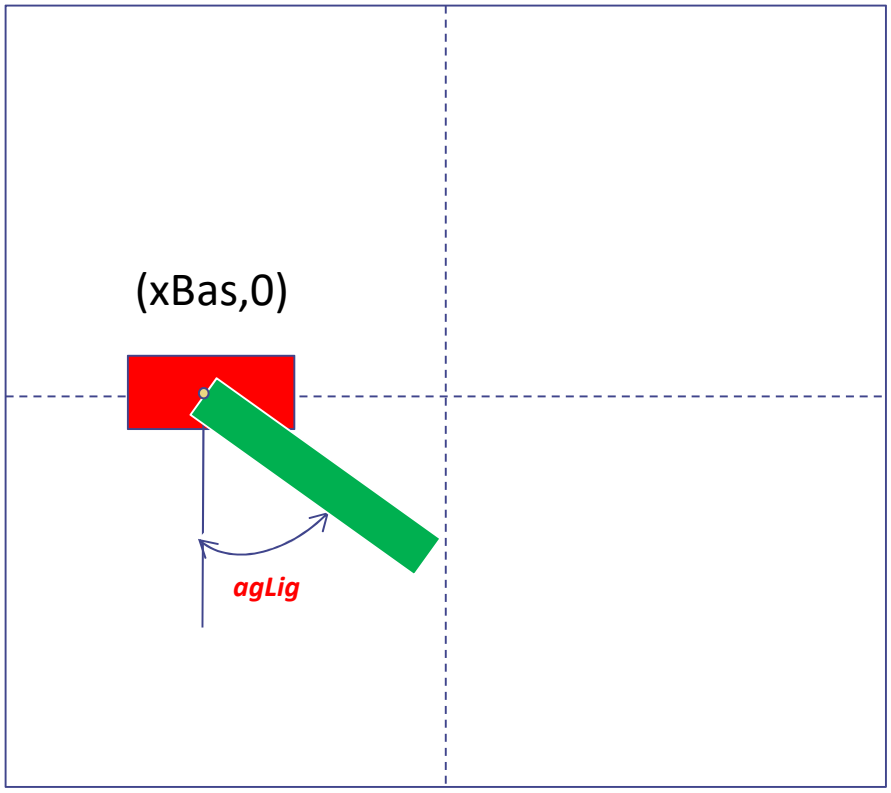
3. Composição de operações

- Transformações comuns / individuais



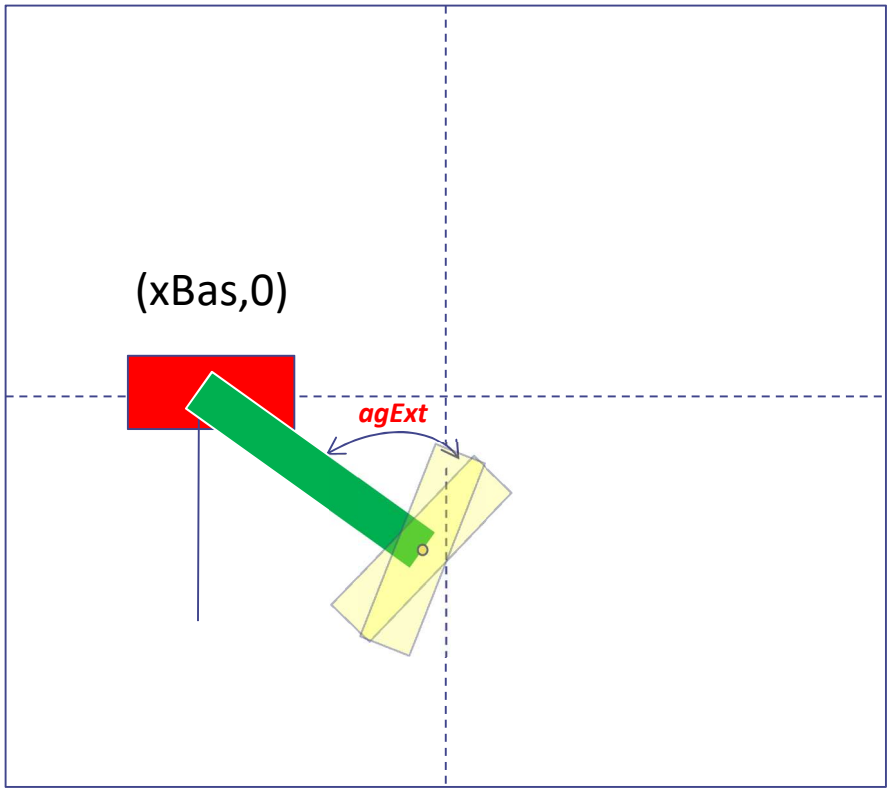

```
glColor3f(1,0,0);
glTranslatef (xSup, 0.0, 0.0);   //GLOBAL a todos os objectos
glPushMatrix();
        glScalef (wSup, hSup, 1.0);
        Quadrado();
glPopMatrix();
```

3. Composição de operações



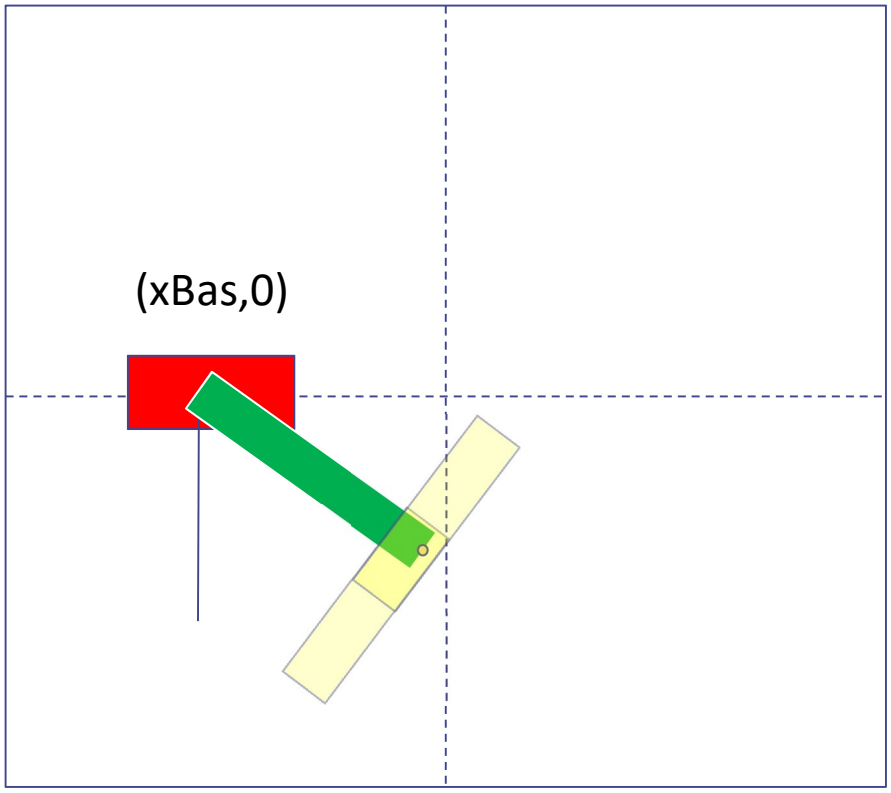

```
glColor3f(0,1,0);
glRotatef (agLig, 0.0, 0.0, 1.0);
glTranslatef ( 0, -hLig/2, 0.0);
glPushMatrix();
        glScalef (wLig, hLig, 1.0);
        Quadrado (1.0);
glPopMatrix();
```

3. Composição de operações



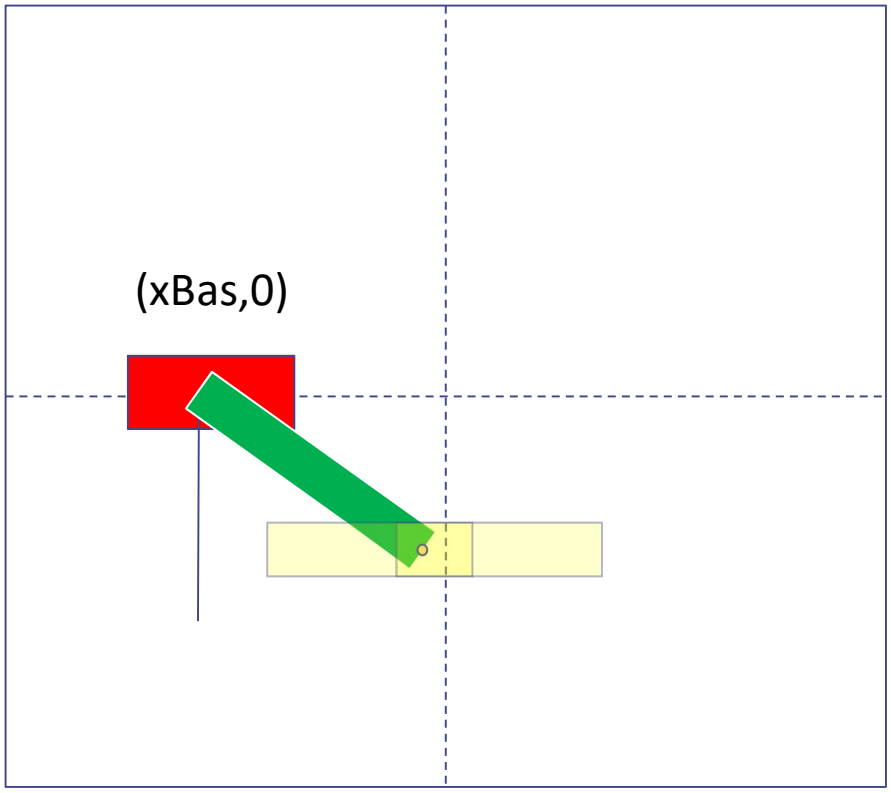

```
glColor3f(1,1,1);
glTranslatef (0, -hLig/2, 0.0);
glRotatef (agExt, 0.0, 0.0, 1.0);
glPushMatrix();
    glScalef (wExt, hExt, 1.0);
    Quadrado (1.0);
glPopMatrix();
```

3. Composição de operações



E se for a mover-se em vez de rodar

3. Composição de operações



E se for a mover-se, mas obrigatoriamente na horizontal ?

- De novo em relação aos trabalhos anteriores
  - A GLUT considera o tratamento de "***teclas***" e "***teclas especiais***" de forma independente

```
main {
        …
            glutKeyboardFunc(teclado);
            glutSpecialFunc(teclasNotAscii);
}
```

```
void teclasNotAscii (int key, int x, int y)
{
            if(key == GLUT_KEY_LEFT) {
                xSup= xSup- incSup;
                glutPostRedisplay();
            }
            if(key == GLUT_KEY_RIGHT) {
                xSup= xSup+ incSup;
                glutPostRedisplay();
            }
…
}
```