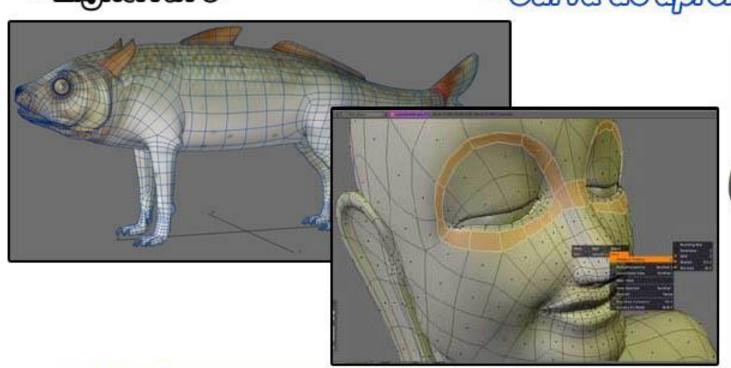


### -CG (Profissional)

- 3DS Max
- Blender
- LightWave



- Licença de uso
- Teanologia
- Curva de aprendizado





- OpenGL (Open Graphical Library)
  - API (Aplication Program Interface)
  - Especificação abenta e multiplatatorma de uma biblioteca de rotinas gráficas e de modelagem utilizada para o desenvolvimento de aplicação de CG

#### 

- -**Jogos** 
  - Sistemas de visualização
    - Visão computacional
      - **E**



#### GLUT

- CLUT ( OpenCL Willity Toolst):
  - Biblioteca destinada para criação de janelas e tratamento de eventos, oferecendo interatividade de forma simples e direta
- Exemplos:
  - entação de janelas e menus pop-up
  - gerenciamento de eventos





- GLU ( OpenGL Willity Library ):
  - Instalada juntamente com a OpenCL
  - Contém uma série de funções que encapsulam comandos OpenGL de batro nível

# código de baixo nível?



- Bibliotecas Necessárias:
  - Instalação prévia das bibliotecas: OpenGL, GLU e GLUT
  - OpenGL: pré-instalada no Windows a partir da versão 93
  - **CLU** e **CLUT:** cópia de **arquivos** DLL ( Dynamic Link Library ) para o subdiretório **System32** do diretório no qual está instalado o Windows





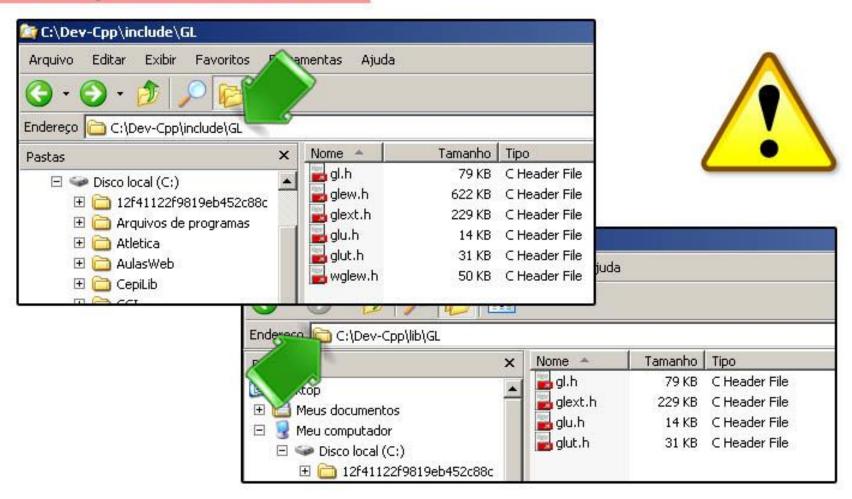
- Instalação no Dev C++
  - Bloodshed:
    http://www.bloodshed.net/dev/devapp.html
  - Download versão: 4992 (?)
    - · IIII@L
    - o Imaludal@L



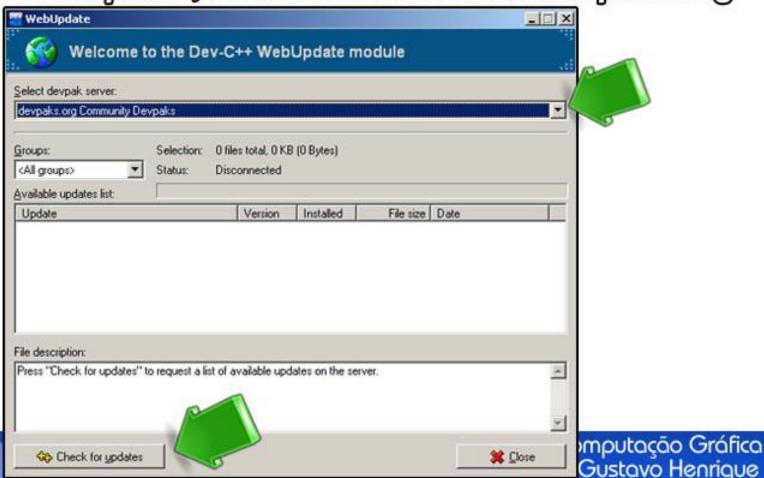
incluir a biblioteca GLVT

- Vertilear no diretário include/GL a existência dos arquivos: gl.h e glu.h

#### - Instalação no Dev C++



- Instalação da Biblioteca GLUT
  - Menu: Ferramentas > Atualizações
  - Na janela WebUpdate, selecione o servidor: devpaks.org



- Bibliotecas Adicionais
  - Necessárias para recursos avançados
  - Libjpeg
    - leitura de imagens Jpeg



- Clew
  - carga automática das extensões OpenGL

- Bibliotecas Adicionais (Dev C++)
  - Libjpeg
    - Menu: Ferramentas > Atualizações

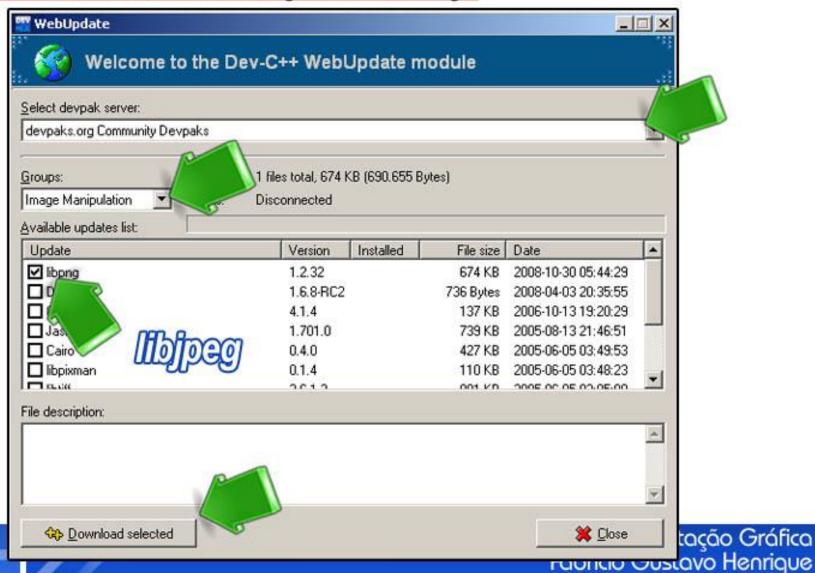


- Download: selecionar o grupo image manipulation
- Selections Ithipper ((download selected))





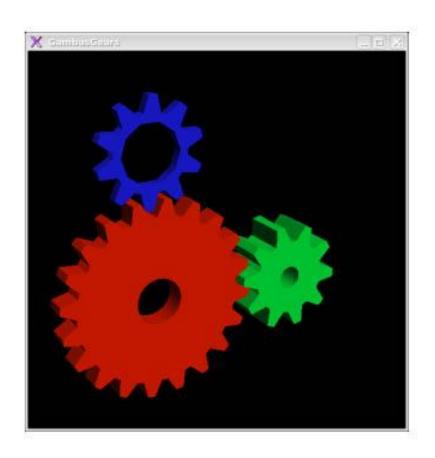
#### - Bibliotecas Adicionais ( Dev C++ )



- Bibliotecas Adicionais (Dev C++)
  - Claw
    - Download da versão atual http://glew.sourceforge.net



- Contetido do arquivo (bim, doc, include e lib)
- -bin Copiar glew\$2.dll para o system\$2 (Windows)
- include/GL Copie todos os arquivos para o diretório correspondente



- Criando um projeto
  - Menu Arquivo > Novo / Projeto
    - Clique na aba Multimedia (feone CLUT)
    - Defina o mome do projeto

- Selecione (Projeto G++)



- Iniciando o OpenGL
  - Objetivo:
    - Desembar um titangulo na janela
    - ESC para sair do programa
  - Carantir a funcionalidade das bibliotecas OpenGL

```
#include <stdlib.h>
#include <GL/glut.h>

// Função callback de redesenho da janela de visualização
void Desenha(void)

{

// Limpa a janela de visualização com a cor branca
glClearColor(1,1,1,0);
glClear(GL_COLOR_BUFFER_BIT);

// Define a cor de desenho: vermelho
glColor3f(0,0,1);
```

- Função Desenha - Limpar tela ( duas tunções ) glclearColor(1,1,1,0); glclear(GL\_COLOR\_BUFFER\_BIT);

```
#include <stdlib.h>
#include <GL/glut.h>

// Função callback de redesenho da janela de visualização
void Desenha(void)

{

// Limpa a janela de visualização com a cor branca
glClearColor(1,1,1,0);
glClear(GL_COLOR_BUFFER_BIT);

// Define a cor de desenho: vermelho
glColor3f(0,0,1);
```

#### - Função Desenha

- Cordo Objeto (desenho) glColor31(Q,Q,1) - RGB;

```
#include <stdlib.h>
#include <GL/glut.h>

// Função callback de redesenho da janela de visualização

void Desenha(void)

{

// Limpa a janela de visualização com a cor branca
glClearColor(1,1,1,0);
glClear(GL_COLOR_BUFFER_BIT);

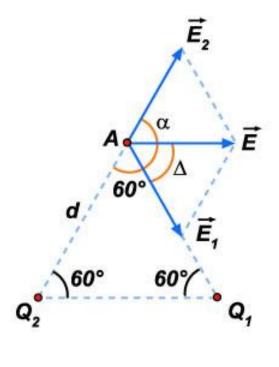
// Define a cor de desenho:
glColor3f(0,0,1);

- 3 parâmetros

- Cores: red, green e blue
```

- Função Desenha - Objeto (triângulo) givertexto(x, y, z);

```
// Desenha um triângulo
glBegin(GL_TRIANGLES);
    // Utilizando 03 parâmetros (pontos)
    glVertex3d(-0.2, -0.2,0);
    glVertex3d(0.0, 0.2,0);
    glVertex3d(0.2, -0.2,0);
glEnd();
```



#### Quantas vértice têm um triangulo?

- Função Tedado
  - Chamada após um determinado evento
  - ASCII
  - -**E**SC(27)

```
// Função callback chamada para gerenciar eventos de teclas
void Teclado (unsigned char key, int x, int y)
{
   if (key == 27)
       exit(0);
}
```

- Função Inicializa
  - Chamada durante a inicialização do programa
  - Determina o tipo e parâmetro de projeção
  - Projeção paralela ortográfica (gluOrtho2D)

```
// Função responsável por inicializar parâmetros e variáveis
void Inicializa(void)
{
    // Define a janela de visualização 2D
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-1.0,1.0,-1.0,1.0);
    glMatrixMode(GL_MODELVIEW);
}
```

Exibição do objeto bidimensional

#### - Programa Principal

return 0;

```
// Programa Principal
int main(void)
                                                                    tunção main
   // Define do modo de operação da GLUT
   glutInitDisplayMode(GLUT SINGLE | GLUT RGB);
   // Especifica o tamanho inicial em pixels da janela GLUT
   glutInitWindowSize(640,480);
   // Cria a janela passando como argumento o título da mesma
   glutCreateWindow("Primeiro Programa");
   // Registra a função callback de redesenho da janela de visualização
   glutDisplayFunc(Desenha);
   // Registra a função callback para tratamento das teclas ASCII
   glutKeyboardFunc (Teclado);
   // Chama a função responsável por fazer as inicializações
   Inicializa();
   // Inicia o processamento e aguarda interações do usuário
   glutMainLoop();
```

