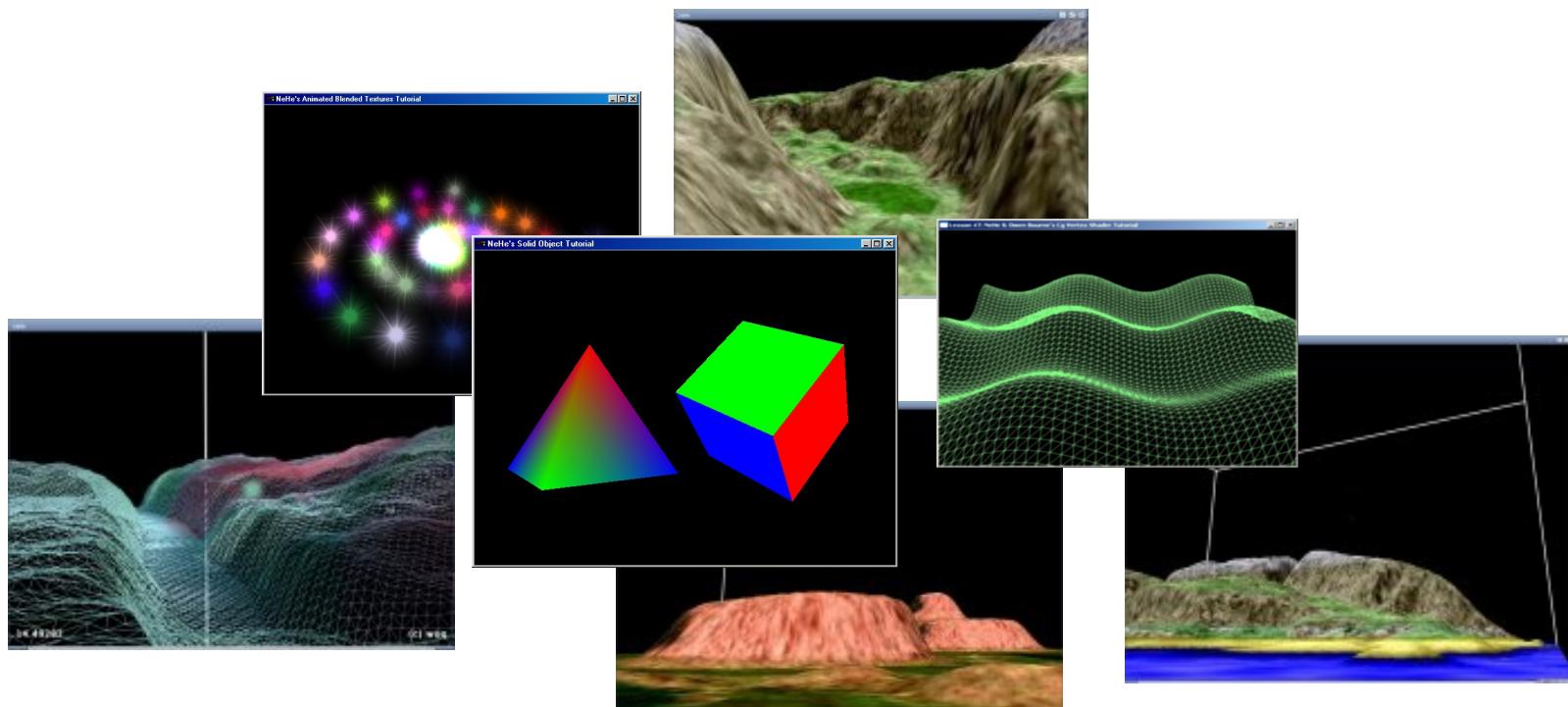


# 3D Graphics Programming

---

Michal Turek  
17.04.2006



# Content

---

- Definitions, Acronyms
- OpenGL
- Coordinate System
- Primitives
- The „Hello, World!“ Example
- Solid Objects
- Transformations
- References
- Questions

# Definitions, Acronyms

---

- OpenGL – Library for 3D Graphics programming
  - MesaGL – Free implementation of OpenGL (SW)
  - Direct3D – Microsoft's library for 3D graphics
- 
- Vertex – The triangle has three vertices
  - Polygon – Triangles, quads, etc.
  - Primitive – The simplest scene object
  - Rendering – „Drawing“ the scene on the screen

# OpenGL Library

---

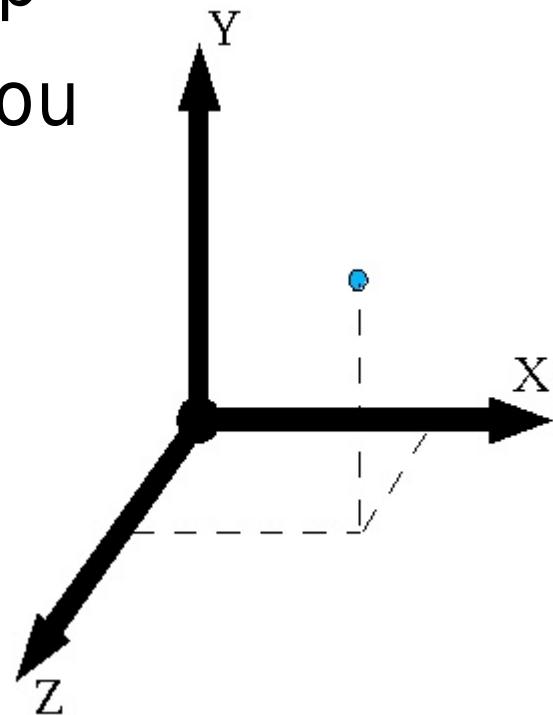
- Industrial standard for 2D and 3D graphics
- Silicon Graphics Inc. (SGI), begining of 1990s
- API between program and video card for CAD applications, virtual reality, games, ...
- Software and hardware implementations
- Supported by most platforms and OS
- New functions of the modern video cards accessible by the „OpenGL Extensions“



# Coordinate System

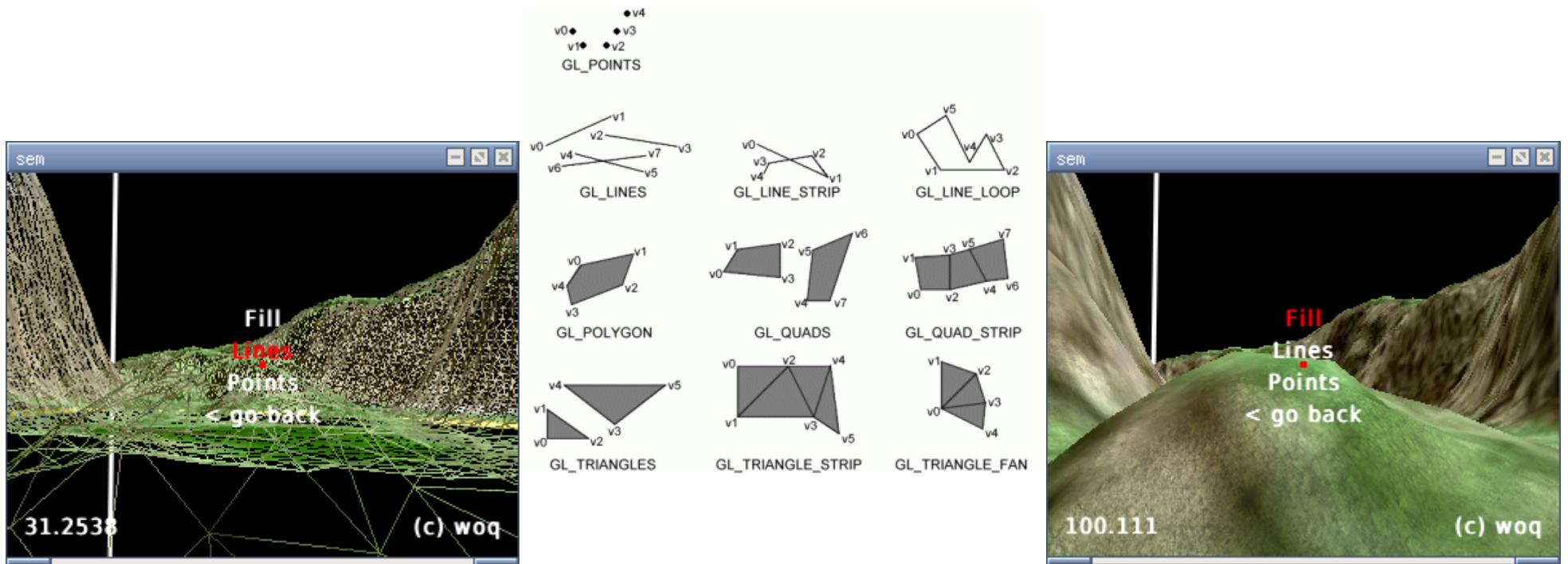
---

- The center of the scene is at [0,0,0]
- X axis goes from left to right
- Y axis goes from bottom to top
- Z axis goes from monitor to you



# Primitives

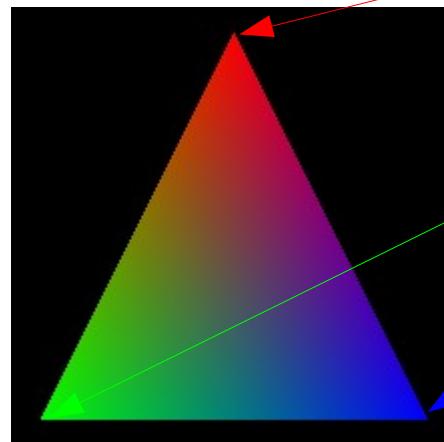
- Everything in the scene consists of primitives like points, lines and polygons



# The „Hello, World!“ Example

---

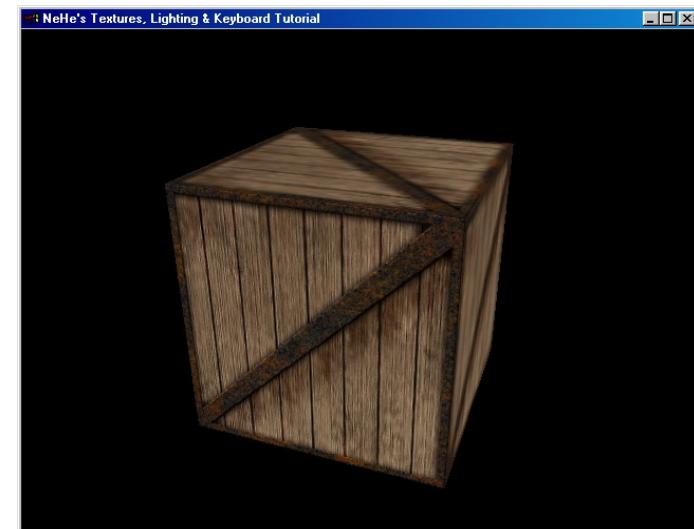
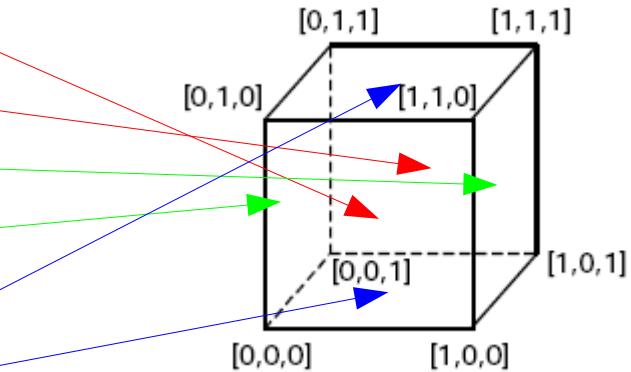
```
void Render()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    glBegin(GL_TRIANGLES);
    glColor3ub(255, 0, 0); glVertex3f( 0.0f, 1.0f, -5.0f); •
    glColor3ub(0, 255, 0); glVertex3f(-1.0f, -1.0f, -5.0f); •
    glColor3ub(0, 0, 255); glVertex3f( 1.0f, -1.0f, -5.0f); •
    glEnd();
}
```



# Solid Objects

```
GLint vertices =  
{  
    0,0,0, 1,0,0, 1,1,0, 0,1,0,  
    0,0,1, 1,0,1, 1,1,1, 0,1,1,  
    1,0,0, 1,0,1, 1,1,1, 1,1,0,  
    0,0,0, 0,0,1, 0,1,1, 0,1,0,  
    0,1,0, 1,1,0, 1,1,1, 0,1,1,  
    0,0,0, 1,0,0, 1,0,1, 0,0,1  
};
```

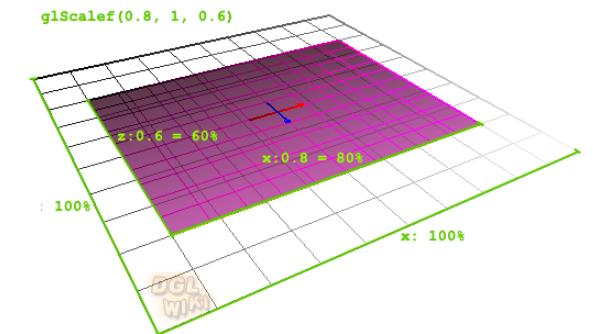
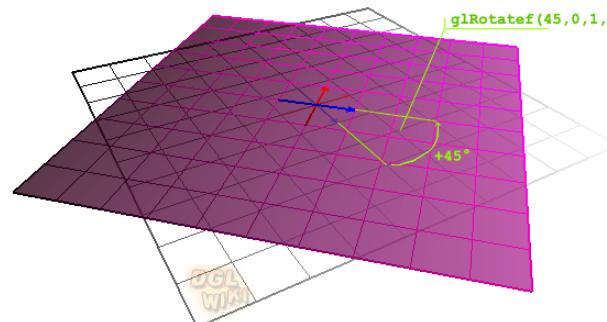
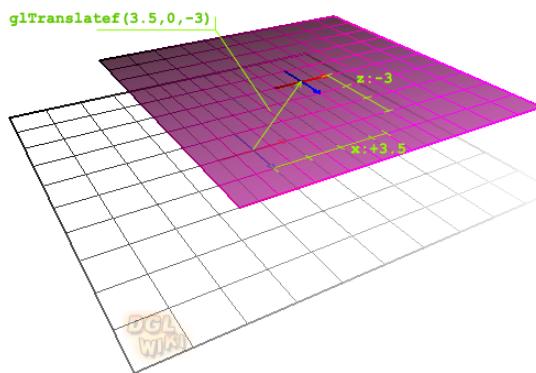
```
glDrawElements(  
    GL_QUADS,  
    24,  
    GL_INT,  
    vertices);
```



# Transformations

---

- Reset – `glLoadIdentity();`
- Translation – `glTranslatef(x, y, z);`
- Rotation – `glRotatef(angle, x, y, z);`
- Scale – `glScalef(x, y, z);`
- `gluLookAt(position, direction, up);`



# References

---

- <http://nehe.gamedev.net/>
- <http://www.gametutorials.com/>
- <http://www.opengl.org/>
- <http://nehe.ceskehry.cz/>
- <http://www.root.cz/> - fulltext „opengl“
- Daniel Čech: OpenGL, Referát na praktikum z informatiky  
([http://nehe.ceskehry.cz/cl\\_gl\\_referat.pdf](http://nehe.ceskehry.cz/cl_gl_referat.pdf))
- Linux man pages, Google

# Questions?

---



# **Thank you...**

---