BIM引领着建设领域的第二次革命

BIM系统开发概论

清华大学土木工程系 胡振中 副教授

huzhenzhong@tsinghua.edu.cn

http://www.huzhenzhong.net

提纲

- 概述
- 软件工程概述
- C#面向对象设计语言简介
- 三维图形平台开发技术简介
- 结语

提纲

- 概述
- 软件工程概述
- C#面向对象设计语言简介
- 三维图形平台开发技术简介
- 结语

概述

- 什么人会用到软件系统？
  - 业务人员编写电子表格程序来简化工作
  - 科学家和工程师编写程序处理试验数据
  - 业余爱好者为了自己的兴趣和爱好编写程序
  - 绝大多数的开发都是专业化的活动
    - 达到特定的业务目的
    - 植入其他设备作为软件产品
    - 例如OA系统、信息管理系统、CAD系统、APP等

概述

- 开发一个软件系统的流程？
- 开发一个软件系统要用到什么技术？
- 开发一个BIM系统与其他软件系统有什么区别？

提纲

- 概述
- 软件工程概述
- C#面向对象设计语言简介
- 三维图形平台开发技术简介
- 结语

软件工程概述

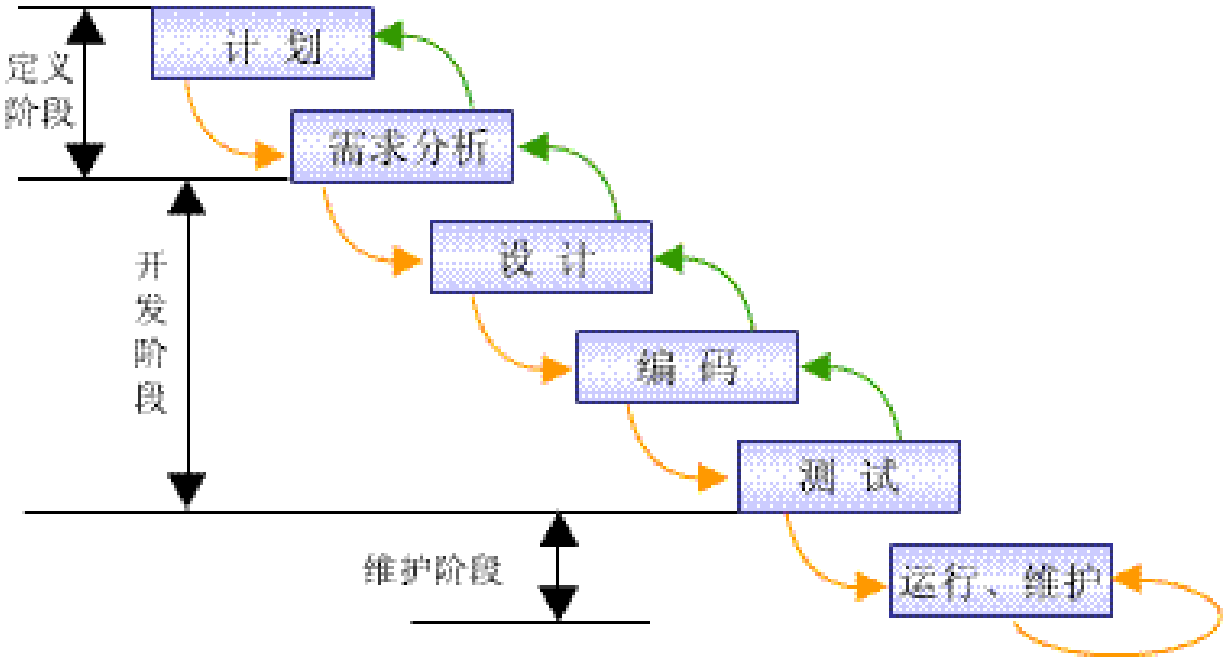
- 软件工程的历史
  - 1968年，一个国际会议，主题是“软件危机”
  - 单个程序的开发技术，无法适应大型、复杂软件系统
  - 提出“软件工程”的概念
  - 20世纪70-80年代，软件工程技术和方法得到发展
    - 结构化编程
    - 信息隐藏
    - 面向对象开发

软件工程概述

- 软件工程的定义和活动
  - 软件工程是关于软件生产的各个方面的工程学科
  - 软件工程中系统化的方法也叫软件过程
  - 所有的软件过程活动都包括4项基本的活动：
    - 软件描述
    - 软件开发
    - 软件验证
    - 软件进化
  - 计算机科学（理论和基础）-- 软件工程（开发和交付）

软件工程概述

- 软件过程模型
  - 软件过程的简化表示，从一个特定的侧面表现软件过程
  - 瀑布模型

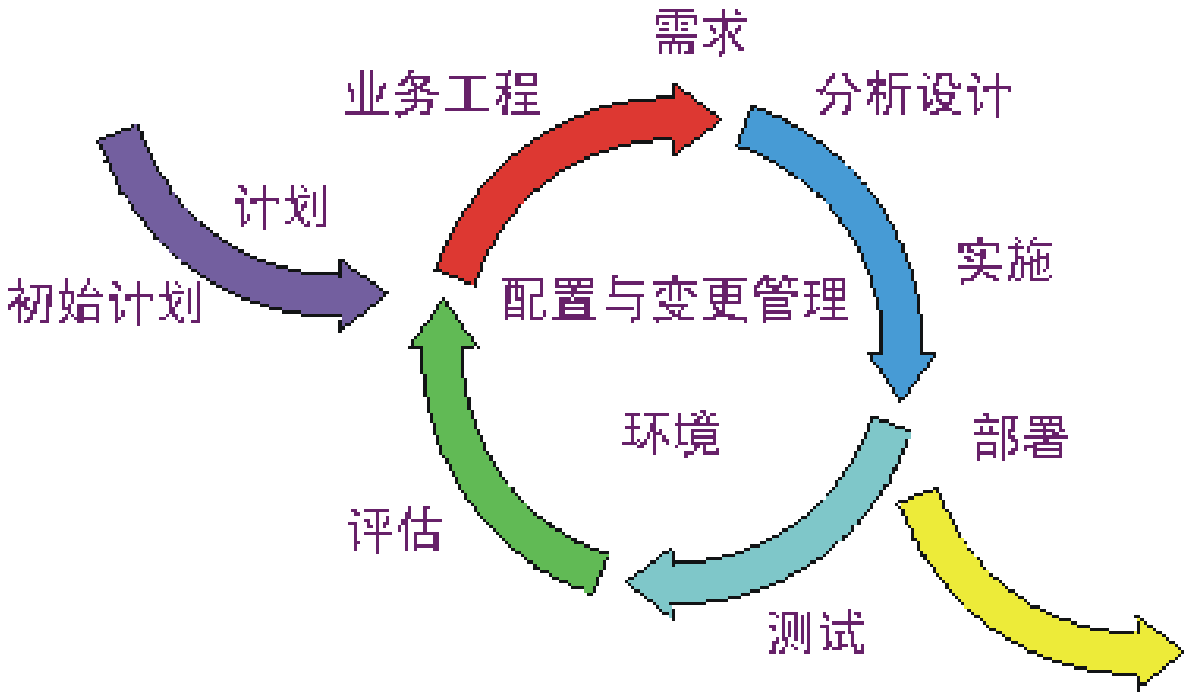


将基本的过程活动看成是一些界限分明的独立的过程阶段

软件工程概述

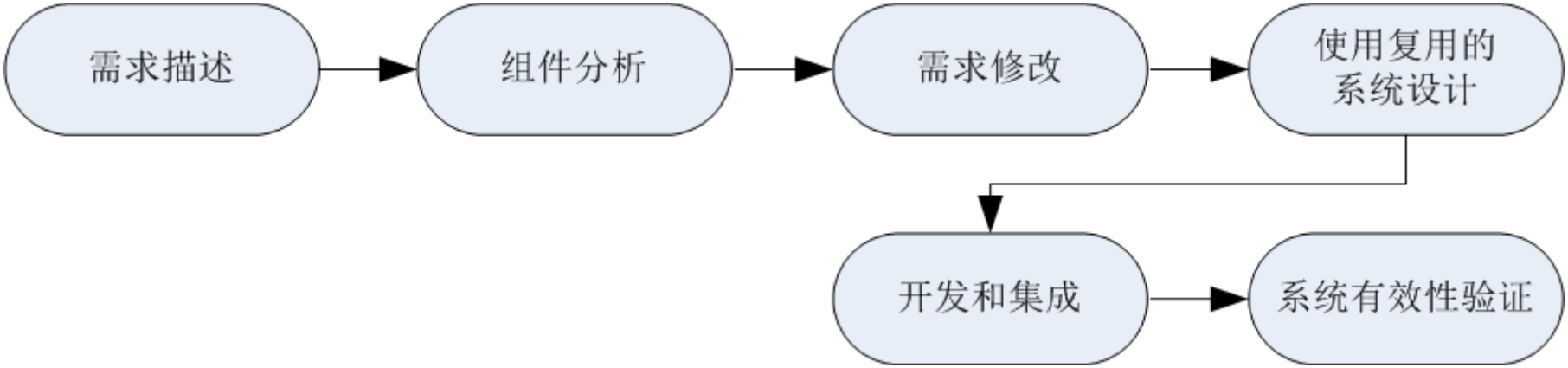
- 软件过程模型
  - 软件过程的简化表示，从一个特定的侧面表现软件过程
  - 增量式开发模型

采用随着日程时间的进展而交错的线性序列，每一个线性序列产生软件的一个可发布的“增量”



软件工程概述

- 软件过程模型
  - 软件过程的简化表示，从一个特定的侧面表现软件过程
  - 面向复用的模型



当项目中的设计或代码与之前的项目有想象的时候，应用其可复用的组件以重新架构

软件工程概述

- 软件过程模型
  - 软件过程的简化表示，从一个特定的侧面表现软件过程
  - 其他的软件过程模型
    - 原型实现模型
    - 快速应用模型
    - 螺旋模型
    - 面向对象模型
    - 面向方面的软件开发模型
    - ……

软件工程-实践者的研究方法 (美)Poger S.Pressman总结为五大类

软件工程概述

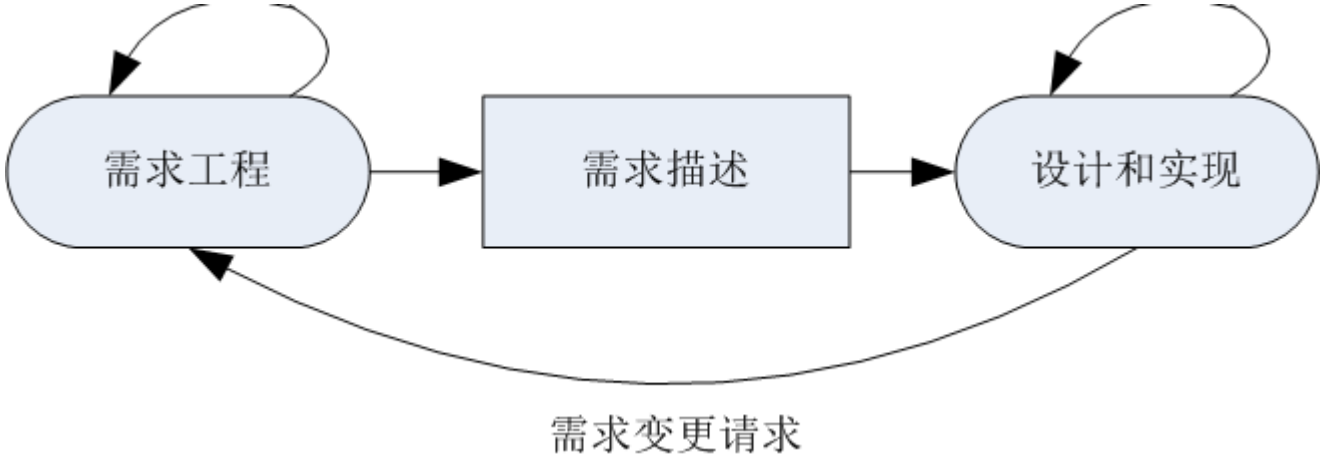
- 使用哪种过程模型？
  - 移动端（Ipad等）查看BIM模型的APP
  - 交互式BIM信息查询系统
  - 一个基于施工BIM信息共享的BIM运维管理系统
  - 轨道交通全生命期BIM平台

软件工程概述

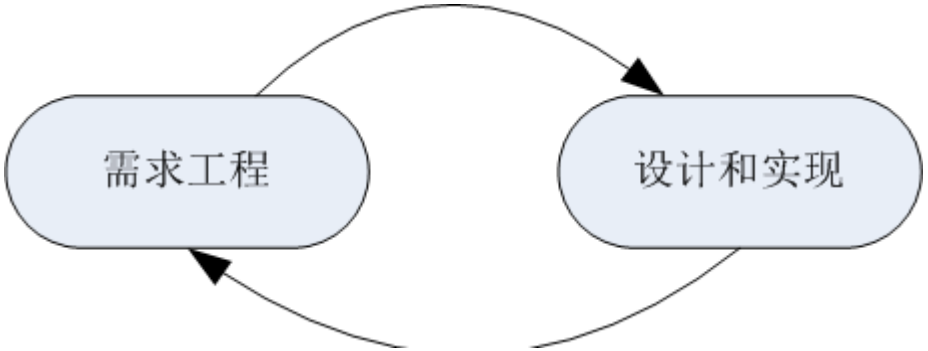
- 计划驱动开发方法和敏捷开发方法
  - 计划驱动开发方法要识别软件过程中的每个阶段及其相关输出，前一阶段的输出作为规划接下来的过程活动的基础
  - 两个问题：迅速的开发，多变的需求
  - 敏捷开发方法认为设计和实现是软件过程的核心活动，是一种专注于快速开发的增量式开发，频繁地发布软件、降低过程开销、生产高质量的代码

软件工程概述

计划驱动开发方法



敏捷开发方法



软件工程概述

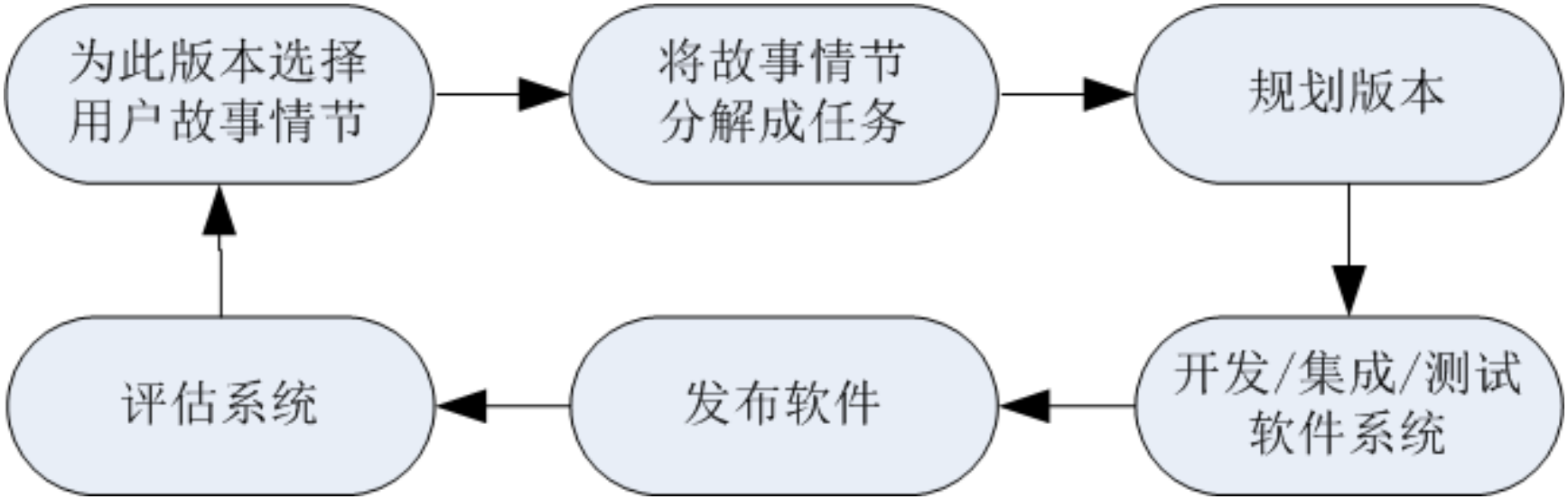
- 选择哪种开发方法？
  - 有非常详细的描述和设计很重要吗？
  - 软件交付给用户后能快速得到反馈吗？
  - 需要大的开发团队来开发的大型系统？
  - 预想的系统寿命长不长？
  - 开发团队是有机组织的还是分散的甚至是外包的？
  - 设计人员和编码人员的能力如何？
  - 系统是否受制于外部的法规？

软件工程概述

- 敏捷软件开发
  - 敏捷方法的基本原则
    - **客户参与**：客户应该在开发过程中始终紧密参与其中。他们的作用是提供新系统的需求，对需求排序，并评估系统的迭代
    - **增量式交付**：以增量的方式开发，客户指定每个增量中的需求
    - **优质团队**：开发团队的技术应得到承认和发扬，团队成员应该保持他们自己的工作风格，不落俗套
    - **接受变更**：预料系统需求的变更，并设计系统使之适应这些变更
    - **保持简单性**：只要可能，就积极地去排除系统中的复杂性

软件工程概述

- 敏捷软件开发之极限编程（XP）
  - 所有的需求都表示为用户故事情节，实现为一系列任务



软件工程概述

- 敏捷软件开发之极限编程（XP）
  - 一个“应急管理”的故事情节，表现为一个“Scenario”

应急管理

小张是一位物业管理人员，负责机场航站楼内给排水管道的维护和应急管理工作。因此，他可以通过点击BIM模型中对应的管道构件，进入构件信息对话框。该对话框中，他可以选择查看“上游构件”、查看“下游构件”、“定位构件”以及“影响范围分析”。

假如他选择“上游构件”，系统会根据与其关联的逻辑关系，列出其上游构件的列表；

假如他选择“下游构件”，系统会根据与其关联的逻辑关系，列出其下游构件的列表；

假如他选择“定位构件”，系统会在三维图形平台中，放大并黄色突出显示该构件，以辅助小张进行控件定位；

假如他选择“影响范围分析”，系统会在三维图形平台中，将该构件对应的所有下游构件放大并红色突出显示。

软件工程概述

- 敏捷软件开发之极限编程（XP）

原则或实践	描述
增量式规则	需求记录在脚本卡片上，开发者将其分解为“任务”
小版本发布	首先开发提供一个最小有用集合，不断地增量式添加功能
简单设计	只进行有限的能满足当前需求的设计，不追求太多
测试优先的开发	在功能本身实现之前，采用自动单元测试框架编写测试
重构	只要发现有可能改善的代码就去做，保持代码简单和可维护

软件工程概述

- 敏捷软件开发之极限编程（XP）

原则或实践	描述
结对编程	开发人员成对工作，检查彼此的工作以支持圆满完成任务
集体所有	配对的开发人员参与系统的所有方面的工作，都有所有权
连续集成	任务一完成，就集成到大系统中，并通过所有的单元测试
可持续节奏	大量超时是不能接受的，其后果通常是降低代码质量和生产率
在场客户	系统客户应该全程配合XP团队，是开发团队的一份子

软件工程概述

- 可扩展的敏捷方法
  - 敏捷方法是为在同一房间办公与交流的小开发团队使用
  - 大型软件系统的大型团队是否适用？
  - Denning认为避免软件工程中不符合客户需求、预算超支等问题的唯一途径是将敏捷方法运用于大型系统
    - 增加设计和系统文档，软件体系必须设计好
    - 建立良好的跨团队沟通机制（wiki，jira）
    - 保持频繁的系统构建和定期发布系统以持续集成（SVN）

提纲

- 概述
- 软件工程概述
- C#面向对象设计语言简介
- 三维图形平台开发技术简介
- 结语

C#面向对象设计语言简介

- 面向对象语言
  - 它设计的出发点就是为了能更加直接地描述客观世界中存在的事件（及对象）以及它们之间的关系
  - 面向对象语言借鉴了20世纪50年代的人工智能语言LISP，引入了**动态绑定**的概念和**交互式开发环境**的思想；始于20世纪60 年代的离散事件模拟语言SIMULA67，引入了**类**和**继承**，成形于20世纪70年代的Smalltalk

C#面向对象设计语言简介

- 面向对象语言
  - 一种是纯面向对象语言，如Smalltalk、EIFFEL等
  - 另一种是**混合型面向对象语言**，即在过程式语言及其它语言中加入类、继承等成分，如C++、C#等

C#面向对象设计语言简介

- 面向对象语言
  - 面向对象的程序设计将现实世界中的客观事物描述成具有属性和行为的对象，通过抽象找出同一类对象的共同属性（静态特征）和行为（动态特征），形成类
  - 类的继承与多态性可以很方便地实现代码重用，大大提高了程序的可重用性，缩短了软件开发周期，并使软件风格统一

C#面向对象设计语言简介

- C#编程语言
  - Microsoft 在 2000 年 7 月推出.NET Framework 时提供的一种全新语言
  - 主要用于基于.NET Framework 的 Windows 和 Web 开发
  - 派生于 C/C++的简洁但高效的语言
  - 又具备Basic易学易用的特点
  - 搭配上Visual Studio等强大的IDE，极大提高了编程效率

C#面向对象设计语言简介

- .NET Framework
  - .NET Framework 包含了一个非常强大的代码库，可以在多种语言（如C#）中通过面向对象编程技 术（OOP）来使用这些代码
  - 使用.NET Framework 编写应用程序，就是使用.NET 代码库编写代码（使用支持 Framework 的任 何一种语言）
  - 最重要的功能之一是垃圾回收（garbage collection），可确保应用程序不再使用某些内存时，就会被完全释放

C#面向对象设计语言简介

- C#编程语言的几种常见应用类型
  - Windows 应用程序：具有我们很熟悉的Windows外观和操作方式，使用.NET Framework的Windows Forms模块就可以简便地生成这种应用程序
  - Web 应用程序：它们是一些Web页面，可以通过任何Web浏览器查看。我们可以使用C#通过Web Forms创建ASP.NET应用程序
  - Web 服务：使用Web服务可以通过Internet虚拟交换数据

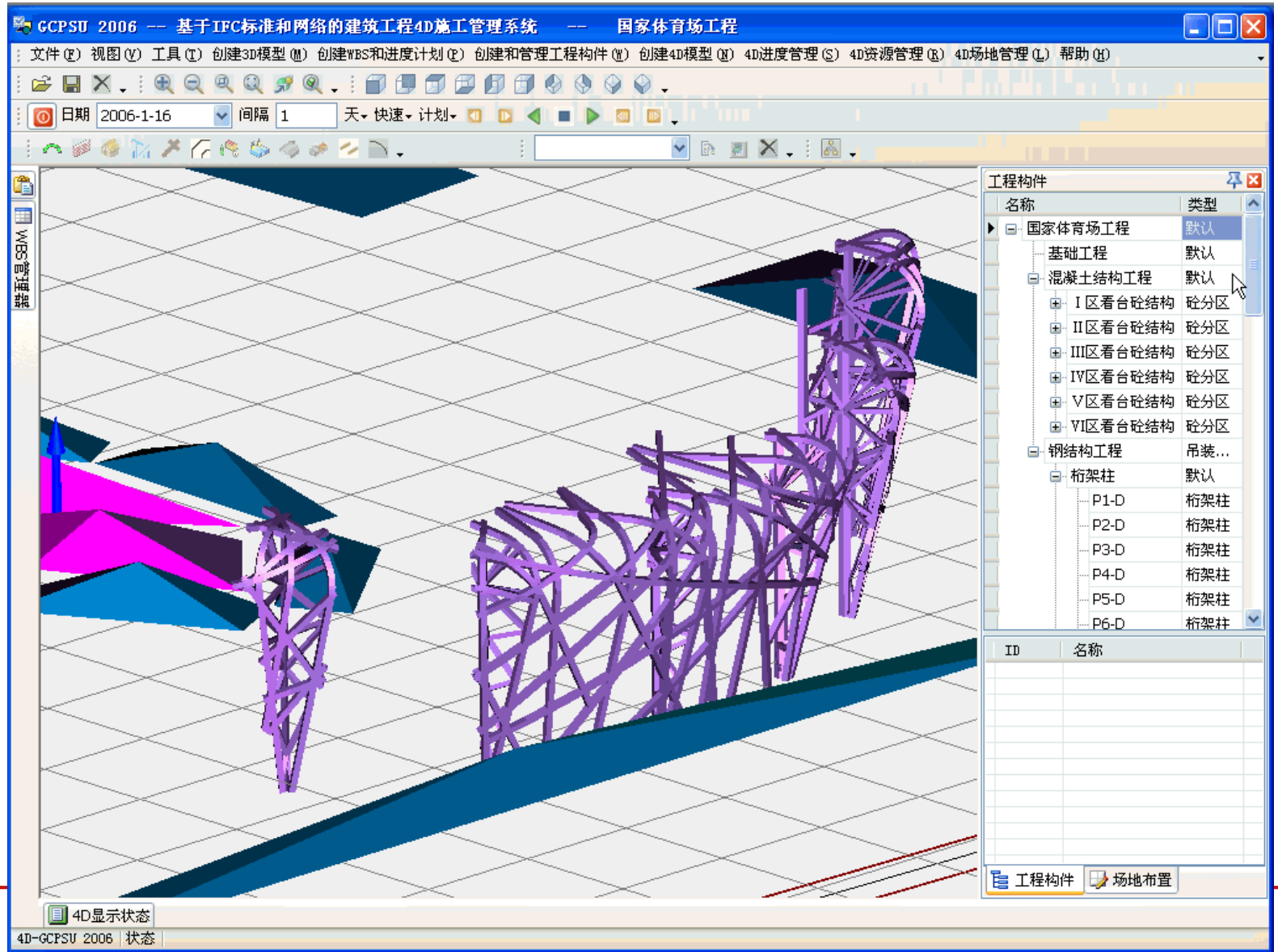
提纲

- 概述
- 软件工程概述
- C#面向对象设计语言简介
- 三维图形平台开发技术简介
- 结语

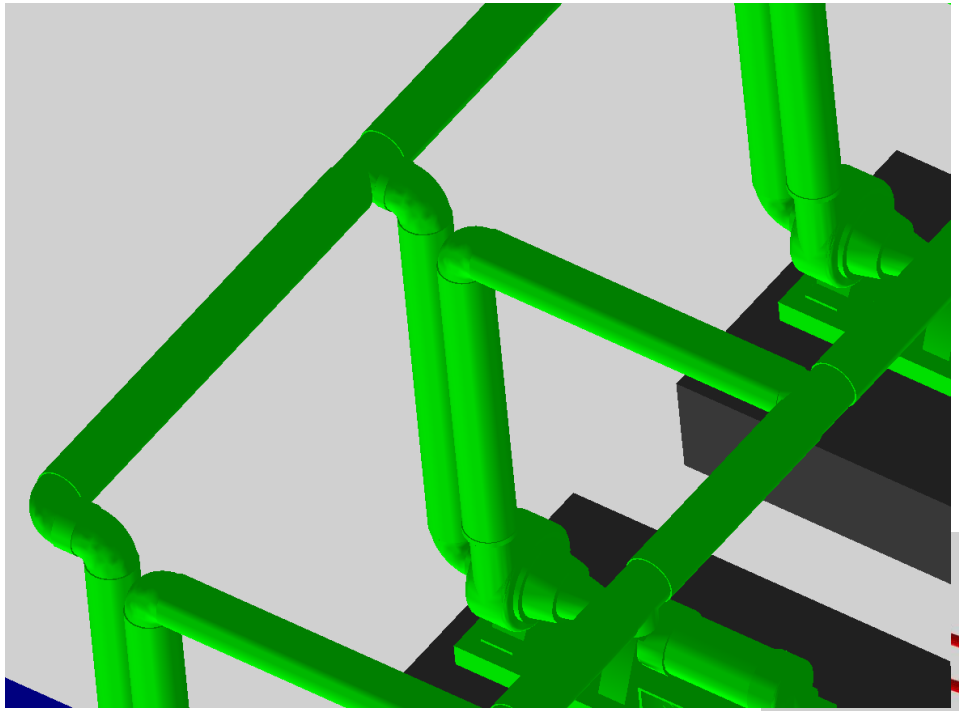
三维图形平台开发技术简介

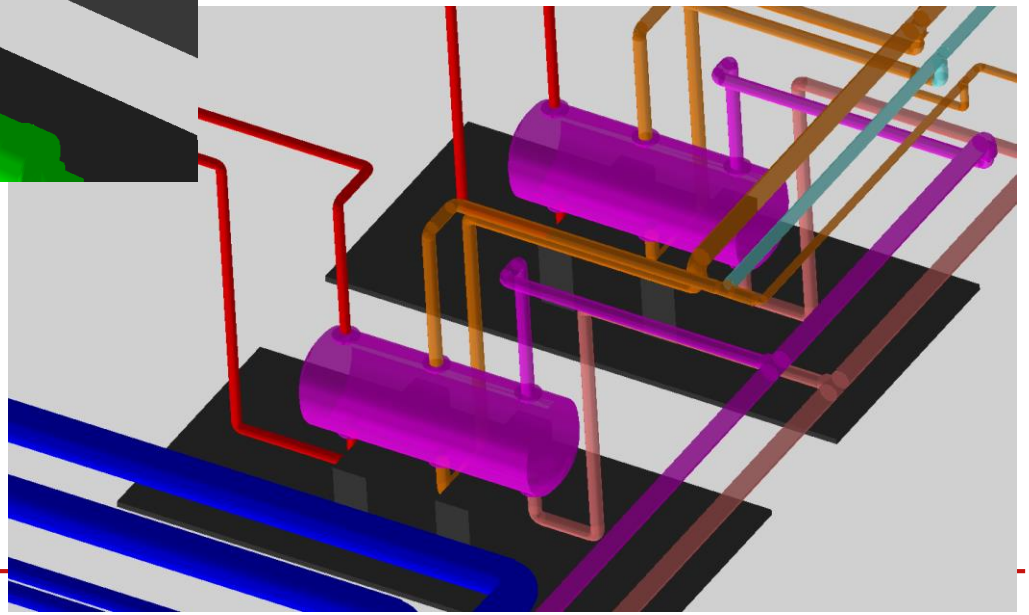
- BIM是基于三维模型的技术和管理理念
- 如何搭建一个BIM图形平台？
  - 基于商品化软件的图形系统
  - 开发具有自主知识产权的三维图形平台
    - 基于操作系统的底层API（OpenGL、DirectX 3D）
    - 基于开源的封装好的操作系统底层API
      - Csgl
      - Three.js
      - ……

三维图形平台开发技术简介

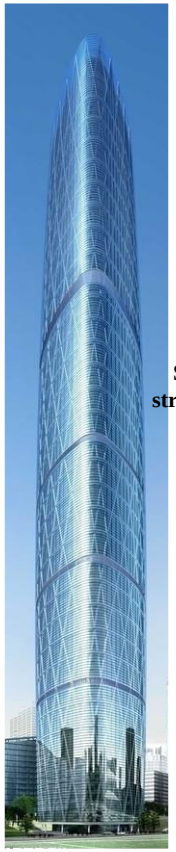


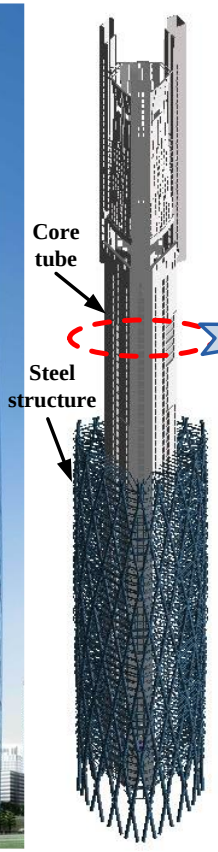
三维图形平台开发技术简介

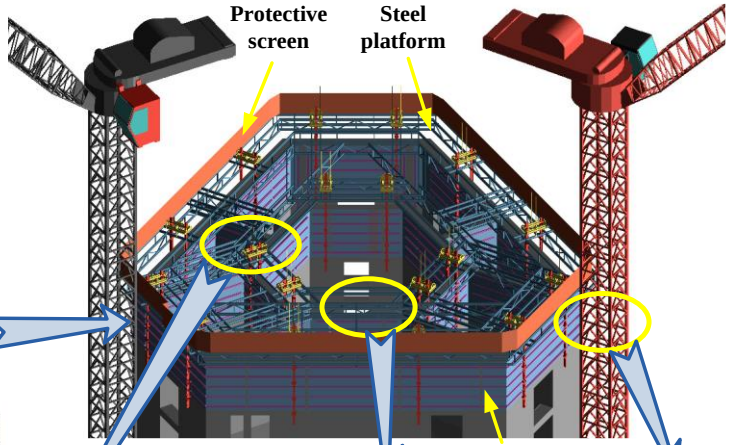


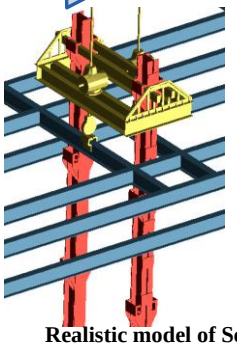


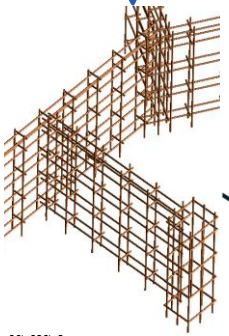
三维图形平台开发技术简介



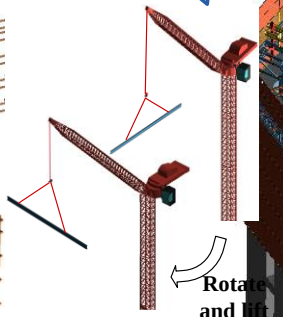




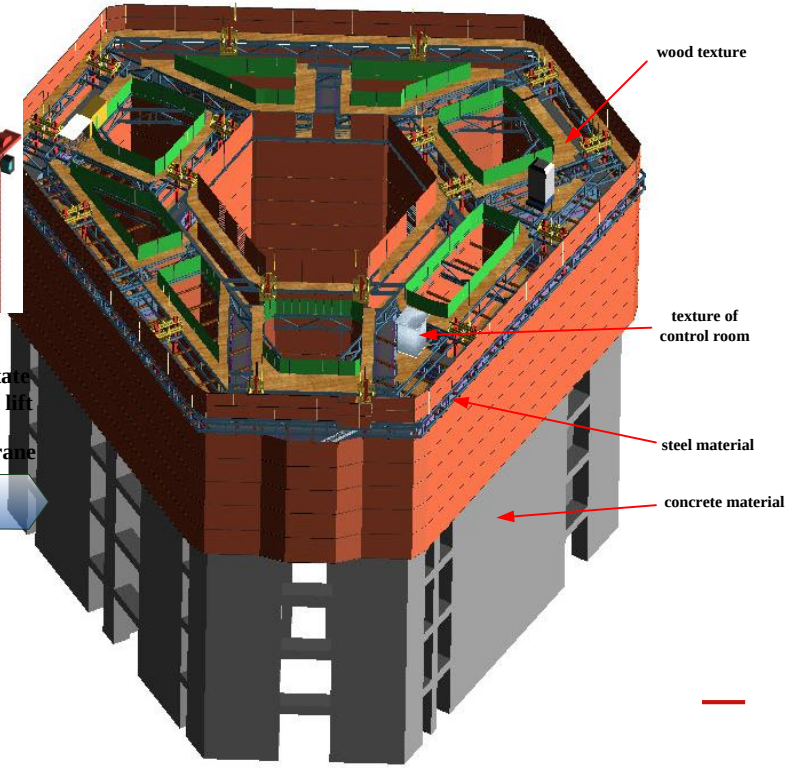




Realistic model of Self-lifting system and reinforcing bars



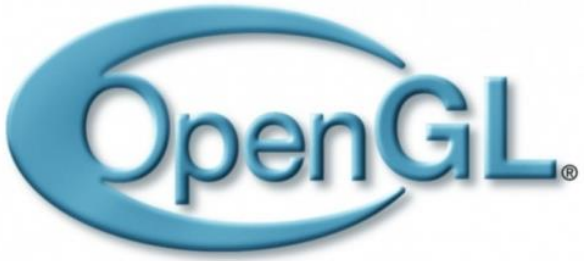
Animation of a tower crane



Design Sketch → Geometric model → Graphic information model for VC

三维图形平台开发技术简介

- OpenGL简介
  - SGI公司推出的OpenGL因性能优越的交互式三维图形建模能力，得到了Microsoft、IBM、DEC、Sun、HP等大公司的认同。因此，OpenGL已经成为一种三维图形开发标准，是从事三维图形开发工作的必要工具
  - 是一个图形应用程序编程接口或函数库



三维图形平台开发技术简介

- OpenGL简介
  - OpenGL被设计成独立于硬件，独立于窗口系统（如Mac OS、UNIX、Windows、Linux等），在运行各种操作系统的各种计算机上都可用，并能在网络环境下以客户/服务器（Client/Server）模式工作
  - 专业图形处理、科学计算等高端应用领域的标准图形库
  - 各种流行的编程语言都可以调用OpenGL中的库函数（如C、C++、C#、Java等）

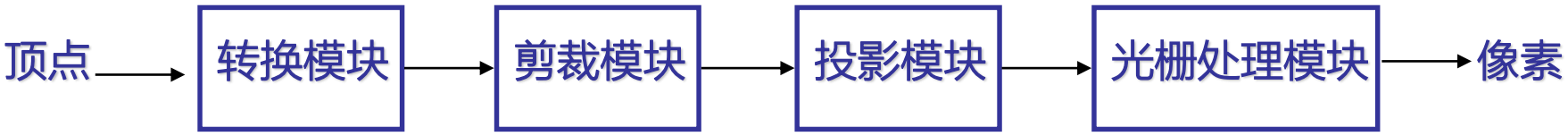
三维图形平台开发技术简介

- OpenGL是什么

从程序员的角度

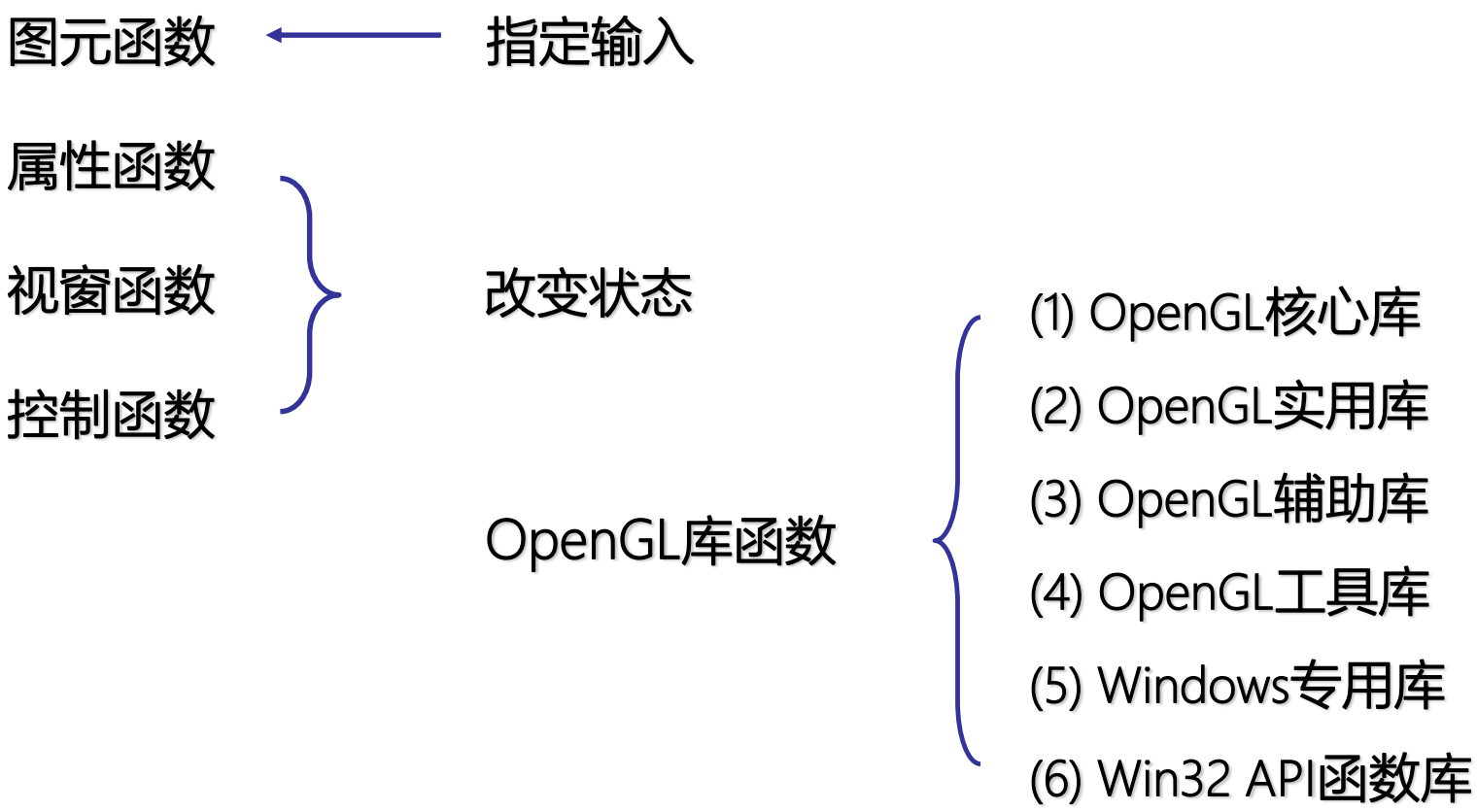
图形应用程序的3要素
- 指定要绘制的对象
- 描述这些对象的属性
- 定义观察这些对象的方式

OpenGL是图形渲染管道



三维图形平台开发技术简介

- OpenGL有什么



三维图形平台开发技术简介

- OpenGL库函数
  - OpenGL核心库，包含有115个函数，函数名的前缀为gl
    - 这部分函数用于常规的、核心的图形处理。由于许多函数可以接收不同数据类型的参数，因此派生出来的函数原形多达300多个
  - OpenGL实用库，包含有43个函数，函数名的前缀为glu
    - 这部分函数通过调用核心库的函数，为开发者提供相对简单的用法，实现一些较为复杂的操作。如：坐标变换、纹理映射、绘制椭球、茶壶等简单多边形。OpenGL中的核心库和实用库可以在所有的OpenGL平台上运行

三维图形平台开发技术简介

- OpenGL库函数
  - OpenGL辅助库，包含有31个函数，函数名前缀为aux
    - 这部分函数提供窗口管理、输入输出处理以及绘制一些简单三维物体。OpenGL中的辅助库不能在所有的OpenGL平台上运行
  - OpenGL工具库，包含大约30多个函数，函数名前缀为glut
    - 这部分函数主要提供基于窗口的工具，如多窗口绘制、空消息和定时器，以及一些绘制较复杂物体的函数。由于glut中的窗口管理函数是不依赖于运行环境的，因此OpenGL 中的工具库可以在所有的OpenGL平台上运行

三维图形平台开发技术简介

- OpenGL库函数
  - Windows专用库，包含有16个函数，函数名前缀为wgl
    - 这部分函数主要用于连接OpenGL和Windows95/NT，以弥补OpenGL在文本方面的不足。Windows专用库只能用于Windows95/98/NT等环境中
  - Win32API函数库，包含有6个函数，函数名无专用前缀
    - 这部分函数主要用于处理像素存储格式和双帧缓存。这6个函数将替换Windows GDI中原有的同样的函数。Win32API函数库只能用于Windows/95/98/NT环境中

三维图形平台开发技术简介

- OpenGL中的数据类型

缩写	数据类型	C	OpenGL
b	8位整数	signed char	GLbyte
ub	8位无符号整数	unsigned char	GLubyte GLboolean
s	16位整数	short	GLshort
us	16位无符号整数	unsigned short	GLushort
i	32位整数	int	GLint GLsizei
ui	32位无符号整数	unsigned int	GLuint GLenum GLbitfield
f	32位浮点数	float	GLfloat GLclampf
d	64位浮点数	double	GLdouble GLclampd
		void	GLvoid

三维图形平台开发技术简介

- OpenGL中函数的多种形式

```
glVertex{234}{sifd}(TYPE coords, ...)
glVertex{234}{sifd}v(TYPE *coords)

GLint i, j;
GLfloat x, y, z, point[3];
                    ...
glVertex2i(i, j);
glVertex2f(x, y);
glVertex3f(x, y, z);
glVertex3fv(point);
```

详见 gl.h中定义

三维图形平台开发技术简介

- OpenGL中的部分常数及其含义

字符	含义
GL_POINTS	绘制单个顶点集
GL_LINES	绘制多组独立的双顶点线段
GL_AMBIENT	设置RGBA模式下的环境光
GL_POSITION	设置光源位置
GL_FLAT	设置平面明暗处理模式
GL_SMOOTH	设置光滑明暗处理模式
……	……

三维图形平台开发技术简介

- OpenGL中的绘图功能

（1）建模功能　　（2）变换功能

（3）颜色模式设置　　（4）光照和材质设置

（5）反走样、融合、雾化　　（6） 位图显示和图像增强

（7）纹理映射　　（8）双缓存动画

三维图形平台开发技术简介

- OpenGL中的绘图功能

（1）建模功能

真实世界里的任何物体都可在计算机中用简单的点、线、多边形描述，OpenGL除了提供基本的点、线、多边形的绘制函数外，还提供了比较复杂的三维物体（如球、锥体、多面体、茶壶等）以及复杂曲线和曲面(如Bezier、Nurbs等曲线和曲面)绘制函数，从而可方便构建虚拟三维世界。

三维图形平台开发技术简介

- OpenGL中的绘图功能

（2）变换功能

无论多复杂的图形都是由基本图元组成并经过一系列变换来实现的。OpenGL的模型变换有平移、旋转、缩放等多种变换。投影变换有透视投影和正交投影两种变换。

（3）颜色模式设置

OpenGL提供两种物体着色模式：

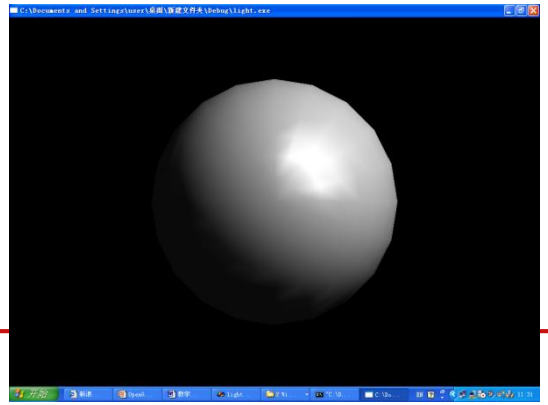
- ✓ RGBA颜色模式
- ✓ 颜色索引模式（Color Index）

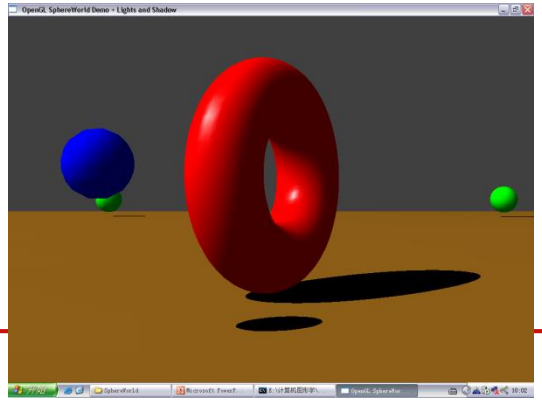
三维图形平台开发技术简介

- OpenGL中的绘图功能

（4）光照和材质设置

OpenGL光源属性有辐射光(Emitted Light)、环境光(Ambient Light)、漫反射光(Diffuse Light)和镜面光(Specular Light)等。材质是用光反射率来表示。场景中物体最终反映到人眼的颜色是光的RGB分量与材质的RGB分量反射率相乘后形成的颜色。



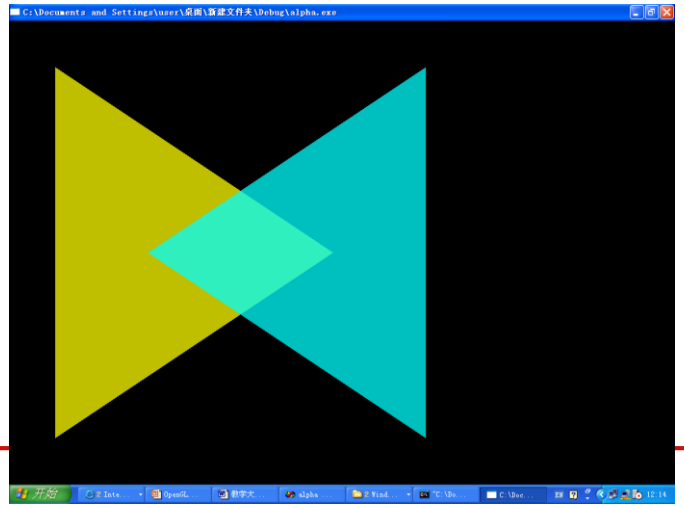


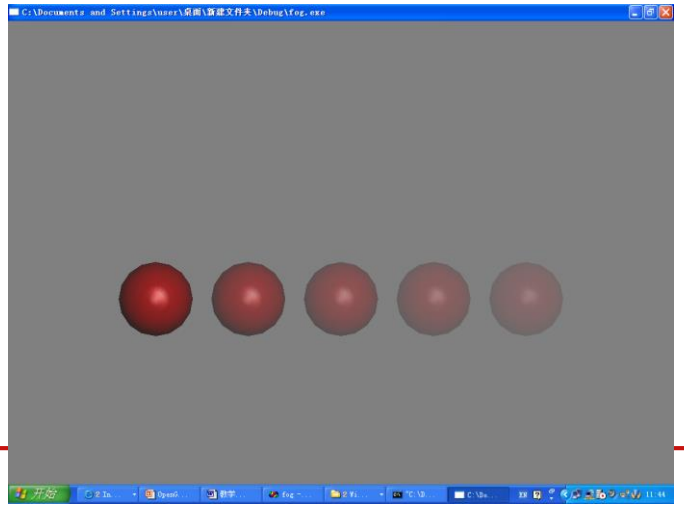
三维图形平台开发技术简介

- OpenGL中的绘图功能

（5）反走样、融合、雾化

OpenGL提供了点、线、多边形的反走样技术。为了使三维图形更加有真实感，经常需要处理半透明或透明的物体图像，这就用到融合技术。OpenGL提供了雾的基本操作来对场景进行雾化处理效果。





三维图形平台开发技术简介

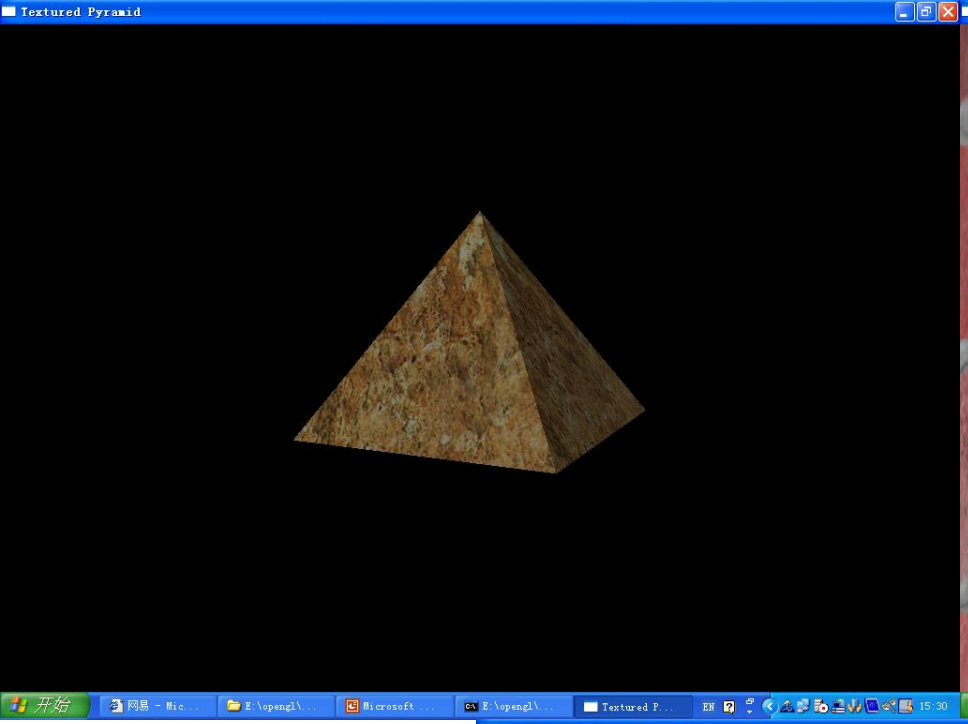
- OpenGL中的绘图功能

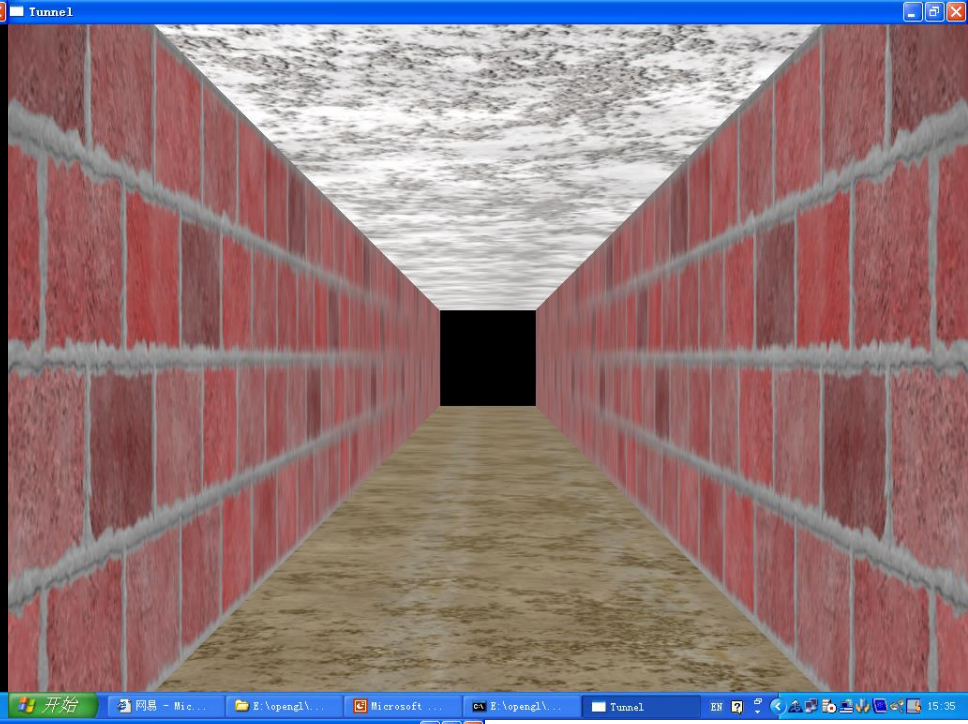
（6）位图显示和图像增强

在图形绘制过程中，图像和位图是非常重要的一个方面，OpenGL提供了一系列函数来实现图像和位图的操作。

（7）纹理映射

在计算机图形学中，将包含颜色、alpha值、亮度等数据的矩形数组称为纹理。而纹理映射可理解为将纹理粘贴在所绘制的三维模型表面，以使三维图形显得更加生动。


Textured Pyramid


Tunnel


OpenGL SphereWorld Demo + Texture Maps

三维图形平台开发技术简介

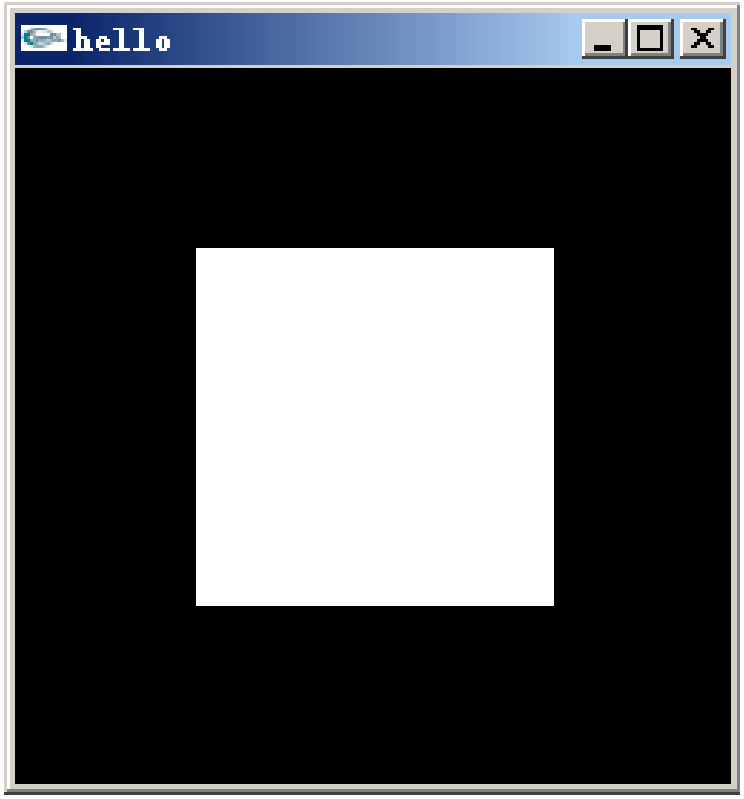
- OpenGL中的绘图功能

（8）双缓存动画

OpenGL提供了双缓存技术来实现动画绘制。双缓存即前台缓存和后台缓存，后台缓存计算场景、生成动画，前台缓存显示后台缓存已画好的画面。

三维图形平台开发技术简介

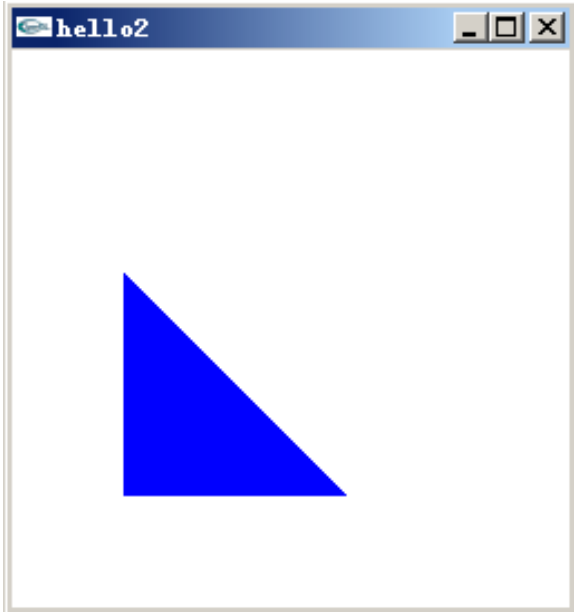
- 程序实例




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

void display(void)
{
        glClear(GL_COLOR_BUFFER_BIT);
        glBegin(GL_POLYGON);
                glVertex2f(-0.5, -0.5);
                glVertex2f(-0.5, 0.5);
                glVertex2f(0.5, 0.5);
                glVertex2f(0.5, -0.5);
        glEnd();
        glFlush();
}

void main(int argc, char * argv[])
{
        glutInit(&argc,argv);
        glutCreateWindow("Simple.C");
        glutDisplayFunc(display);
        glutMainLoop();
}
```

三维图形平台开发技术简介

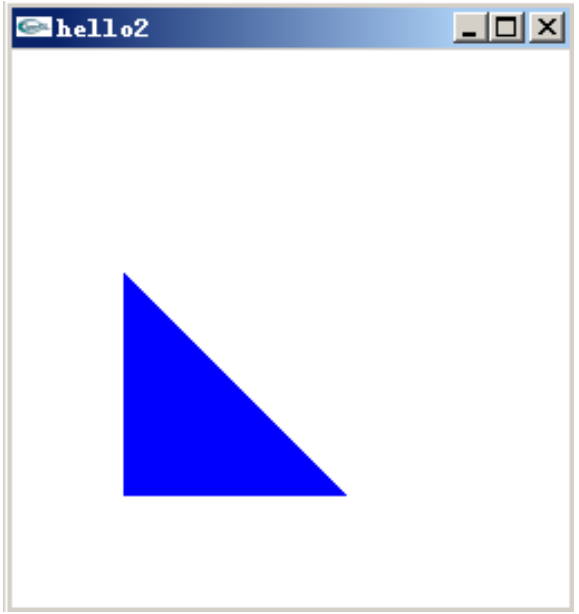
- 程序实例




```
/*
 * hello2.c
 * This is a simple flat color draw blue tirangle | OpenGL program.
 */
#define GLUT_DISABLE_ATEXIT_HACK
#include <GL/glut.h>


void display(void)
{
/* clear all pixels  */
   glClear (GL_COLOR_BUFFER_BIT);
   glShadeModel(GL_FLAT);

/* draw Blue triangle  with corners at
 * (0.2,0.2) , (0.6, 0.2) and (0.2,0.6)
 */

   glBegin(GL_TRIANGLES);
        glColor3f(0.0, 0.0, 1.0);
        glVertex2f(0.2, 0.2);
        glVertex2f(0.6, 0.2);
        glVertex2f(0.2, 0.6);
   glEnd();

/* don't wait!
 * start processing buffered OpenGL routines
 */
   glFlush ();
}
```

三维图形平台开发技术简介

- 程序实例

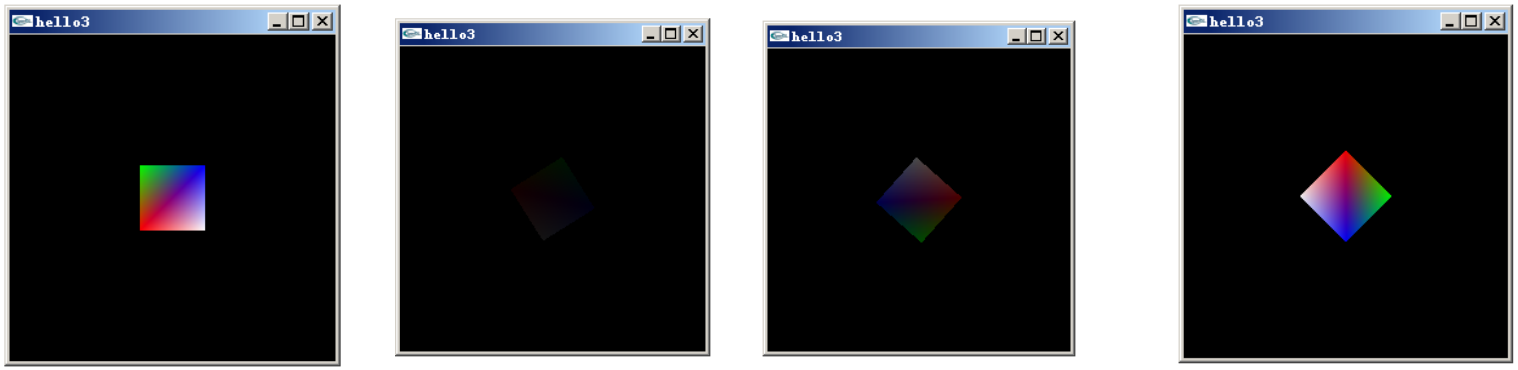



```
void init (void)
{
/* select clearing color    */
   glClearColor (1.0, 1.0, 1.0, 0.0);

/* initialize viewing values  */
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);
}

/*
 * Declare initial window size, position, and display mode
 * (single buffer and RGBA).  Open window with "hello2"
 * in its title bar.  Call initialization routines.
 * Register callback function to display graphics.
 * Enter main loop and process events.
 */
int main(int argc, char** argv)
{
   glutInit(&argc, argv);
   glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
   glutInitWindowSize (250, 250);
   glutInitWindowPosition (100, 100);
   glutCreateWindow ("hello2");
   init ();
   glutDisplayFunc(display);
   glutMainLoop();

}
```

三维图形平台开发技术简介

- 程序实例

绘制一个黑窗口，并显示一个彩色的平滑模式绘制四边形，鼠标左键点击后，目标会进行旋转，右键点击旋转停止。此外，各个点随着旋转，颜色保持不变，深浅要变化。



开始

旋转过程出现深浅变化

右键点击旋转停止
颜色变化也停止

三维图形平台开发技术简介

- 程序实例

```
/*
 *  double.c
 *  This is a simple double buffered program.
 *  Pressing the left mouse button rotates the rect.
 *  Pressing the right mouse button stops the rotation.
 */
#define GLUT_DISABLE_ATEXIT_HACK
#include <GL/glut.h>
#include <stdlib.h>

static GLfloat spin = 0.0;
static GLfloat chagecolor=1.0;

void display(void)
{
   glClear(GL_COLOR_BUFFER_BIT);
   glPushMatrix();
   glRotatef(spin, 0.0, 0.0, 1.0);

   glBegin(GL_POLYGON);
        glColor3f(chagecolor, 0.0, 0.0);
        glVertex2f(-10.0,-10.0);
        glColor3f(0.0,chagecolor, 0.0);
        glVertex2f(-10.0, 10.0);
        glColor3f(0.0, 0.0,chagecolor);
        glVertex2f(10.0, 10.0);
        glColor3f(chagecolor, chagecolor, chagecolor);
        glVertex2f(10.0, -10.0);
    glEnd();

   glPopMatrix();

   glutSwapBuffers();
}
```

三维图形平台开发技术简介

- 程序实例

```
void spinDisplay(void)
{
   spin = spin + 0.02;
   if (spin > 360.0)
      spin = spin - 360.0;
   chagecolor=chagecolor+0.01;
   if (chagecolor > 1.0)
      chagecolor = chagecolor - 1.0;
   glutPostRedisplay();

}

void init(void)
{
   glClearColor (0.0, 0.0, 0.0, 0.0);
   glShadeModel (GL_SMOOTH);
}
```

```
void reshape(int w, int h)
{
   glViewport (0, 0, (GLsizei) w, (GLsizei) h);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   glOrtho(-50.0, 50.0, -50.0, 50.0, -1.0, 1.0);
   glMatrixMode(GL_MODELVIEW);
   glLoadIdentity();
}

void mouse(int button, int state, int x, int y)
{
   switch (button) {
      case GLUT_LEFT_BUTTON:
         if (state == GLUT_DOWN)
            glutIdleFunc(spinDisplay);
         break;
      case GLUT_MIDDLE_BUTTON:
      case GLUT_RIGHT_BUTTON:
         if (state == GLUT_DOWN)
            glutIdleFunc(NULL);
         break;
      default:
         break;
   }
}
```

三维图形平台开发技术简介

- 程序实例

```
/*
 *  Request double buffer display mode.
 *  Register mouse input callback functions
 */
int main(int argc, char** argv)
{
   glutInit(&argc, argv);
   glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
   glutInitWindowSize (250, 250);
   glutInitWindowPosition (100, 100);
   glutCreateWindow ("hello3");
   init ();
   glutDisplayFunc(display);
   glutReshapeFunc(reshape);
   glutMouseFunc(mouse);
   glutMainLoop();

}
```

提纲

- 概述
- 传统的运维管理
- 基于BIM的运维管理技术
- BIM运维的发展趋势
- 结语

结语

- 知识层面
  - 掌握如何系统性地进行软件系统开发
  - 了解进行BIM系统开发的一些基本常用技能和工具
  - 了解三维图形平台的开发技巧
- 科学方法论层面
  - 用工程管理的角度去思考软件工程
  - 创新的三种方式（发现、发明、吸收改进）
  - 打造中国自己的高端平台

谢谢！

清华大学土木工程系
胡振中 副教授
Email: huzhenzhong@tsinghua.edu.cn
个人网站: http://www.huzhenzhong.net