OpenGL图形编程

授课教师：少书师夜

位图图像

- 图像

BMP文件是一种像素文件，它保存了一幅图象中所有的像素。这种文件格式可以保存单色位图、16色或256色索引模式像素图、24位真彩色图象，每种模式 种单一像素的大小分别为1/8字节，1/2字节，1字节和3字节。目前最常见的是256色BMP和24位色BMP。这种文件格式还定义了像素保存的几种方 法，包括不压缩、RLE压缩等。常见的BMP文件大多是不压缩的。

这里为了简单起见，我们仅讨论24位色、不使用压缩的BMP。（如果你使用Windows自带的画图程序，很容易绘制出一个符合以上要求的BMP）

位图图像

- 位图与图像的区别
  - 位图的每个像素仅包含一位信息，图像的每个像素一般含有多种信息（R、G、B、Alpha）；
  - 位图用于掩码，遮盖别的图像；图像数据则覆盖先前数据或与先前数据混合。

位图

- 当前光栅位置

```
void glRasterPos{234}{sifd}[v](TYPE x,TYPE y,TYPE z,TYPE w);
```

- 显示

```
void glBitmap(GLsize width,GLsizei height, GLfloat xbo,GLfloat ybo,GLfloat xbi,GLfloat ybi,const GLubyte* bitmap);
```

像素读写

- 我们可以从BMP文件中读取像素数据，并使用glDrawPixels绘制到屏幕上
  - 读

```
void glReadPixels(GLint x,GLint y,GLsizei width,GLsizei height,GLenum format,GLenum type,GLvoid* pixels);
```

  - 写

```
void glDrawPixels(GLsizei width,GLsizei height, GLenum format,GLenum type,GLvoid* pixels);
```

图像

- 像素复制

```
void glCopyPixels(GLint x,GLint y,GLsizei width,GLsizei height,GLenum buffer);
```

- 图像缩放

```
void glPixelZoom(GLfloat zoomx,GLfloat zoomy);
```

OpenGL消隐

- 多边形剔除

主要用于去除多边形物体本身的不可见面，以提高图形系统的性能。

```
glEnable(GL_CULL_FACE);
glCullFace (mode);
```

参数mode可取值为GL_FRONT、GL_BACK或GL_FRONT_AND_BACK，分别表示剔除多边形的前面、后面或前后面。

深度测试

采用深度缓存器算法，消除场景中的不可见面。

- 深度值设置

默认情况下，深度值的范围在0.0~1.0。可用以下函数修改

```
glDepthRange (nearNormDepth, farNormalDepth);
```

将深度值设置在nearNormDepth到farNormalDepth之间。 nearNormDepth和farNormalDepth可取0.0~1.0的任意值。

深度测试

- 初始化

```
glClearDepth (maxDepth);
glClear(GL_DEPTH_BUFFER_BIT);
```

参数maxDepth取值在0.0~1.0之间。

- 启用与关闭

```
glEnable(GL_DEPTH_TEST);
glDisable(GL_DEPTH_TEST);
```

深度测试

- 测试方式改变

默认情况下，将需要绘制的新像素的z值与深度缓冲区中对应位置的z值相比较，如果比深度缓存中的值小，用新像素的颜色值更新帧缓存中对应的颜色值。但这种方式也可以改变。

```
glDepthFunc(func);
```

参数func的值可为GL_NEVER、GL_ALWAYS、GL_LESS、GL_LEQUAL、GL_EQUAL、GL_GEQUAL、GL_GREATER、GL_NOTEQUAL，默认值为GL_LESS。

雾与显示列表

- 雾
- 显示列表
- 交互

雾

- 雾效果

  随着视点距离的增大，物体变得愈来愈模糊。

1.启用与关闭雾：

- 启用 glEnable(GL_FOG)
- 关闭 glDisable(GL_FOG)

雾

2.设置雾属性

void glFog{if}[v](GLenum pname,TYPE param);

雾效混合因子f

- GL_EXP
- GL_EXP2
- GL_LINEAR

雾

3.雾坐标

- Z：视点与片元之间的距离。默认情况下，Z自动计算产生。
- 也可设置顶点的雾坐标

```
void glFogCoord{fd}[v](TYPE z);
```

雾

- 例子

显示列表

- 适用场合
  - 矩阵操作
  - 光栅位图与图像
  - 光照、材质和光照模型
  - 纹理

显示列表

- 显示列表的创建

```
void glNewList(GLuint listID,GLenum listMode);
```

参数listID为一个不为0的正整数索引值。

参数listMode取值GL_COMPILE或GL_COMPILE_AND_EXECUTE。

显示列表

- 显示列表的创建

如：

```
glNewList( listID, listMode );
glutSolidCube(2.0);
……
glEndList();
```

显示列表

- 显示列表的执行
  - 单个列表执行

```
void glCallList(GLuint listID);
```

  - 多个列表执行

```
void glCallLists(GLsizei n,GLenum
  type,const GLvoid *lists);
```

显示列表

- 多级显示列表

OpenGL支持创建多级显示列表，即在glNewList和glEndLsit函数对之间允许调用glCallList函数来执行其他显示列表。

- 显示列表的删除

```
void glDeleteLists(GLuint listID, GLsizei range);
```

交互

- 橡皮筋技术
- 拾取（选择）
- 菜单

OpenGL实现橡皮筋技术

- 橡皮筋技术的实现方法
  - 利用颜色的异或操作，对原有图形并不是擦除，而是再绘制一条同样的直线段并与原图形进行异或操作，此时原图形会从屏幕上消失；
  - 利用双缓存技术，绘制图形时分别绘制到两个缓存，交替显示。

OpenGL实现橡皮筋技术

- 1.鼠标实现
  - 鼠标响应注册函数

    glutMouseFunc(MousePlot);
  - 鼠标响应函数

  void MousePlot(Glint button,Glint action,Glint xMouse,Glint yMouse)；

  参数button：GLUT_LEFT_BUTTON、GLUT_MIDDLE_BUTTON或GLUT_RIGHT_BUTTON。

  参数action：GLUT_DOWN或GLUT_UP。

  坐标（xMouse,yMouse)制定当前鼠标相对于窗口左上角点的位置坐标。

OpenGL实现橡皮筋技术

- 1.鼠标实现
  - 鼠标移动注册函数

```
glutMotionFunc(MouseMove);
glutPassiveMotionFunc(PassiveMouseMove);
```

  - 鼠标移动相应函数

```
void MouseMove(Glint xMouse,Glint yMouse);
Void PassiveMouseMove(Glint xMouse,Glint yMouse);
```

OpenGL实现橡皮筋技术

- 2.键盘实现
  - 键盘注册函数

```
glutKeyboardFunc(Key);
```

  - 键盘相应函数

```
void Key(unsigned char key,int x,int y);
```

OpenGL实现拾取操作

- 设置拾取缓冲区

```
void glSelectBuffer(GLsizei n, GLunint
  *buff);
```

- 进入选择模式

```
GLint glRenderMode(GLenum mode);
```

OpenGL实现拾取操作

- 名字堆栈操作
  - 初始化名字堆栈（glInitNames）
  - 将一个名字压入堆栈（glPushName）
  - 替换名字堆栈的栈顶元素（glLoadName）
  - 将栈顶元素弹出（glPopName）

OpenGL实现拾取操作

- 设置合适的变换过程

```
gluPickMatrix(xPick, yPick, widthPick, heightPick, *vp);
```

- 为每个图元分配名字并绘制
- 切换回渲染模式
- 分析选择缓冲区中的数据

OpenGL中的菜单功能

- 菜单注册函数

  glutCreateMenu(ProcessMenu);

- 在菜单中加入菜单项

  void glutAddMenuEntry(char *name, GLint value);

- 将菜单与某个鼠标按键关联

  void glutAttachMenu(button);