**《计算机图形学》实验报告一**

专业:数字媒体技术 班级: 22中德媒体1 学号: 2022705114

姓名: 张文杰

|  |  |  |  |
| --- | --- | --- | --- |
| 实验名称 | OpenGL的简单动画 机器时钟动画 | | |
| 实验地点 | 家庭个人电脑 | 实验时间 | 1 |
| 1.实验目的：  1.掌握OpenGL的闲置函数。  2.掌握OpenGL的时间函数。  3.掌握OpenGL的简单动画功能。  4.了解OpengGL裁剪窗口、视区、显示窗口的概念和它们之间的关系。  5.进一步掌握OpenGL的基本图元的绘制。  2实验内容（原理和方法）  **1.闲置函数的使用与简单动画。**  1) 旋转的六边形，如图2-1所示。  阅读6.3.3节中旋转的六边形样本框架程序，分析程序的实现步骤。运行该程序，观察旋转动画效果。  思考: 如果要调整旋转速度，旋转更快或更慢，应该如何修改程序？  答：修改旋转角增量,即通过更改myidle（）闲置函数的属性值theta大小实现    2） 线框六边形。  在display 函数中，添加多边形模式设置语句观看效果。  glPolygonMode(GL\_FRONT\_AND\_BACK,GL\_LINE); //线框模式  添加线宽语句观看效果。    glLineWidth(2.0); //设置线宽  重回多边形填充模式：  glPolygonMode(GL\_FRONT\_AND\_BACK,GL\_FILL); //填充模式  3) 在图形中添加字符"Hello"，观察结果；然后将"Hello"字符改为自己名字的拼音或英文名字。  提示：在图形中添加如下代码：  glColor3f(1,0,0);  //设置红色绘制颜色  glRasterPos2i(30,20);    //定位当前光标，起始字符位置  glutBitmapCharacter(GLUT\_BITMAP\_9\_BY\_15,'H');  //写字符"H"  glutBitmapCharacter(GLUT\_BITMAP\_9\_BY\_15,'e');  //写字符"e"  glutBitmapCharacter(GLUT\_BITMAP\_9\_BY\_15,'l');   //写字符"l"  glutBitmapCharacter(GLUT\_BITMAP\_9\_BY\_15,'l');   //写字符"l"  glutBitmapCharacter(GLUT\_BITMAP\_9\_BY\_15,'o');   //写字符"o"  4) 变色技术举例：  在程序头部设置全部变量:  int k=0;  在myidle函数中添加代码：  if (k==1)  {  glColor3f(1,0,0) ;  k=0;  }  else  {  glColor3f(1,1,0) ;  k=1;  }  然后在绘制函数中屏蔽原来的绘制颜色，运行查看效果。  效果就是：红色和黄色交替变化，很闪。    5) 六边形静止，直线单独旋转。  修改前面的程序，使得六边形保持静止，以六边形中心为起点画一条不同颜色的直线，终点为六边形某一顶点，使得直线不停绕中心点旋转。代码保存下来备用。思考：如果需要直线保持与机器时钟的秒针节拍吻合，应该如何修改？  答：要直线保持与机器时钟的秒针节拍吻合，可在myidle函数中设置Sleep(1000)延时1秒，并设置theta += 6\*PI/180，使得每秒转过6°，一分钟转一周。  提示：可用延时 Sleep()函数,如Sleep(1000)表示延时1秒，放在 myidle函数中。  **2.时间函数的使用与简单动画。**  将以上程序中的闲置函数替换为时间函数，  1)主程序中的glutIdleFunc(myidle);    //注册闲置回调函数  改为:  glutTimerFunc(1000, mytime,10); //1000毫秒后调用时间函数 mytime  2)myidle()闲置回调函数改为时间函数 mytime(t)在程序顶部，函数声明语句也要相应更改：  void myidle();  改为:  void mytime(int t);  3)在时间函数 mytime(int t)最后再添加:  glutTimerFunc(1000, mytime,10); //1000毫秒后调用时间函数 mytime    3.简单时钟的设计。  1)在程序头部定义系统时间变量，时分秒变量：  SYSTEMTIME timeNow;  float hh,mm,ss;  2)在程序头部定义Π常量:  #define PI 3.1415926  3）在程序头部引入数学头文件、时间头文件：  #include "math.h"  #include "time.h"  4）在初始化函数中获取系统时间:  在主程序中顶部声明初始化子函数  void init();  在main函数中 添加子函数调用语句，可放在创建窗口之后：  init();  在main函数后面，添加初始化子函数，并在函数中添加获取系统时间语句：  void init()  {      GetLocalTime(&timeNow);    //获取系统时间      hh=timeNow.wHour;    //获取小时时间  mm=timeNow.wMinute;   //获取分钟时间  ss=timeNow.wSecond;      //获取秒时间  }    5）在绘制函数中计算时分秒，确定绘制时分秒针起始点坐标，例如：  //xc,yc为时针中心点坐标  //xs,ys为秒针终止点坐标  //xm,ym为分针终止点坐标  xs=xc+R\*cos(PI/2.0-ss/60\*2\*PI);  ys=yc+R\*sin(PI/2.0-ss/60\*2\*PI);  xm=xc+R\*cos(PI/2.0-(mm+ss/60.0)/60.0\*2.0\*PI);  ym=yc+R\*sin(PI/2.0-(mm+ss/60.0)/60.0\*2.0\*PI);  xh=xc+(R-5)\*cos(PI/2.0-(hh+(mm+ss/60.0)/60.0)/12.0\*2.0\*PI);  yh=yc+(R-5)\*sin(PI/2.0-(hh+(mm+ss/60.0)/60.0)/12.0\*2.0\*PI);    6）在绘制函数中以直线方式简易绘制时分秒针  glColor3f(1,0,0);  glBegin(GL\_LINES);  glVertex2f(xc,yc);  glVertex2f(xs,ys);  glEnd();    glColor3f(1,1,0);  glBegin(GL\_LINES);  glVertex2f(xc,yc);  glVertex2f(xm,ym);  glEnd();    glColor3f(0,1,1);  glBegin(GL\_LINES);  glVertex2f(xc,yc);  glVertex2f(xh,yh);  glEnd();    7）闲置函数中 或 时间函数中重复获取系统时间  GetLocalTime(&timeNow);    //获取系统时间  hh=timeNow.wHour;    //获取小时时间  mm=timeNow.wMinute;   //获取分钟时间  ss=timeNow.wSecond;      //获取秒时间  3流程图       1. 源程序 2. 源文件保存在我自己的GitHub仓库中   <https://github.com/frankswsdbob/JSUT_OpenGL_Exercise.git>   1. 效果截图和演示视频 | | | |