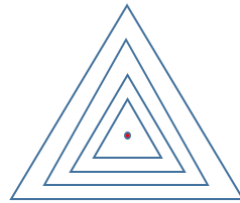


高斌_大作业 1 设计说明书

一、 作业要求

大作业1： 动画等边三角形

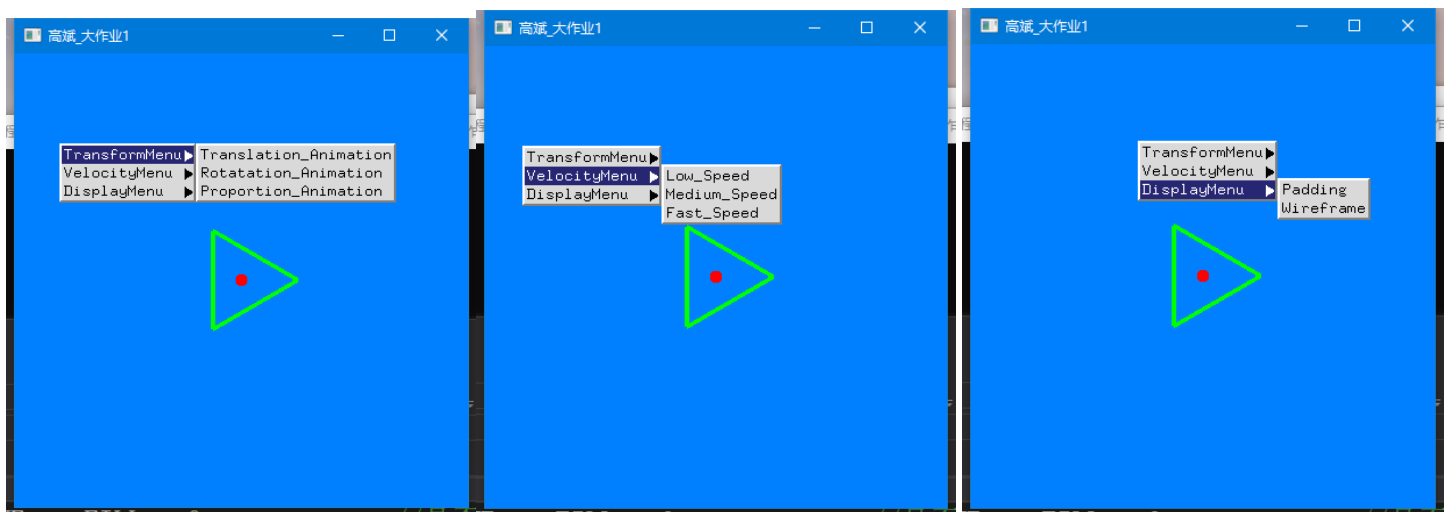
- 颜色、大小、初始位置都自定，右键弹出菜单，左键选择菜单
- 变换（一级菜单）
 - 平移动画（二级菜单）：从左端移动到右端，再从右端移动到左端。匀速
 - 旋转动画（二级菜单）：绕几何中心顺时针旋转一圈，再逆时针旋转。匀速
 - 比例动画（二级菜单）：相对几何中心，先放大比例到某个最大值，再缩小回到原始大小。匀速
- 速度（一级菜单）
 - 慢速（二级菜单）
 - 中速（二级菜单）
 - 快速（二级菜单）
- 显示（一级菜单）
 - 填充（二级菜单）
 - 线框（二级菜单）



注：动画不能满屏全是三角形，该图仅仅示意三角形几何中心以及比例变化。

二、 操作

- 1、打开“发布版_直接运行”文件夹，双击 exe 文件，然后右键弹出菜单；
- 2、分别是一级菜单变换、速度和显示，一级菜单下有二级菜单，左键点击即可播放动画。（因为中文显示是乱码，菜单中用英文代替中文）



```
PS E:\vcpkg> .\Windows PowerShell
PS E:\vcpkg> ./cpgk install freeglut:x64-windows
Computing installation plan...
The following packages will be built and installed:
    freeglut[core]:x64-windows -> 3.2.145
Detecting compiler hash for triplet x64-windows...
-- Automatically setting HTTP(S) PROXY environment variables to localhost:10800
Could not locate cached archive: C:\Users\gaobin712\AppData\Local\vcpkg\archives\5f5f24d0496f0d3a05c863fd23f37511c194114d56.zip
Starting package 1/1: freeglut:x64-windows
building package freeglut[core]:x64-windows...
-- Downloading https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=cfhable,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=pilotfiber,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=sigenet,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=versaweb,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=ayera,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=netactuate,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=phoenixap,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=autismisnet,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=freafz,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=netcologne,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=deac-riga,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=iweb,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=jalist,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=jztktft,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=managedway,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=netix,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=ufpr,https://sourceforge.net/projects/freetgl/files/freetgl/3.2.1/freetgl-3.2.1.tar.gz/download?use_mirror=tenet -> freeglut-3.2.1.tar.gz...
-- Extracting source E:\vcpkg\downloads\freetgl-3.2.1.tar.gz
-- Applying patch use_targets_to_export_xll_dependency.patch
-- Applying patch macOS_Quantz.patch
-- Applying patch gcc10_patch
-- Applying patch fix-debug-macro.patch
-- Applying patch no_x64_enforcement.patch
-- Using source at E:\vcpkg\buildtrees\freetgl/src/aced4dbcd3-ea0f90c226.clean
-- Configuring x64-windows
-- Building x64-windows-dbg
-- Building x64-windows-rel
-- Installing: E:\vcpkg\packages\freetgl_x64-windows/share/freetgl/copyright
-- Performing post-build validation
-- Performing post-build validation done
Stored binary cache: C:\Users\gaobin712\AppData\Local\vcpkg\archives\5f5f24d0496f0d3a05c863fd23f37511c194114d56.zip
Building package freeglut[core]:x64-windows... done
Installing package freeglut[core]:x64-windows... done
Elapsed time for package freeglut:x64-windows: 27.59 s

Total elapsed time: 31.69 s

The package freeglut is compatible with built-in CMake targets:

    find_package(GLUT REQUIRED)
    target_link_libraries(main PRIVATE GLUT::GLUT)
```

3、将 vcpkg 安装的库集成到 Visual studio2017

```
PS E:\vcpkg> .\vcpkg integrate install
Applied user-wide integration for this vcpkg root.

All MSBuild C++ projects can now #include any installed libraries.
Linking will be handled automatically.
Installing new libraries will make them instantly available.

CMake projects should use: "-DCMAKE_TOOLCHAIN_FILE=E:\vcpkg\scripts\buildsystems\vcpkg.cmake"
PS E:\vcpkg>
```

4、打开 Visual studio2017 创建控制台项目就可以直接调用 OpenGL 的库了，无需再配置。

2、项目实施流程：

(1) 首先在初始化窗口。

```
/*初始化，创建窗口*/
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
glutInitWindowPosition(400, 150);
glutInitWindowSize(800, 800);
glutCreateWindow("高斌_大作业1");
```

(2) 用 glutCreateMenu 函数创建一级菜单，然后在一级菜单下用 glutAddMenuEntry 加入二级菜单。下面是“变换菜单的例子”：

```
/*变换一级菜单及子菜单*/
int gaobin_TransformMenu = glutCreateMenu(TransformMenu);
glutAddMenuEntry("Translation_Animation", 1);
glutAddMenuEntry("Rotatation_Animation", 2);
glutAddMenuEntry("Proportion_Animation", 3);
```

(3) 在创建完成后，将一级菜单集成到主菜单中，并用 glutAttachMenu(GLUT_RIGHT_BUTTON) 设置为右键弹出菜单。

```
/*主菜单*/
int gaobin_MainMenu = glutCreateMenu(MianMenu);
glutAddSubMenu("TransformMenu", gaobin_TransformMenu);
glutAddSubMenu("VelocityMenu", gaobin_VelocityMenu);
glutAddSubMenu("DisplayMenu", gaobin_DisplayMenu);
glutAttachMenu(GLUT_RIGHT_BUTTON); //右键弹出菜单
```

(4) 然后就是主要的绘制函数 void gaobin_Triangle(void)，在设置完基础的参数后，绘制一个等边三角形；

```
/*三角形*/
glLineWidth(4.0f); // 设置线的宽度
glBegin(GL_TRIANGLES);
    glColor3d(0.0, 1.0, 0.0); //设置颜色
    glVertex2d(0.25*cos(0), 0.25*sin(0));
    glVertex2d(0.25*cos(240 * PI / 180), 0.25*sin(240 * PI / 180));
    glVertex2d(0.25*cos(120 * PI / 180), 0.25*sin(120 * PI / 180));
glEnd();
```

(5) 菜单设置完成之后，我们点击菜单传给菜单函数的是设置的值，我们将菜单中的平移量、旋转量、缩放的值、速度等级和是否填充设为全局变量。然后在绘制函数 `void gaobin_Triangle(void)` 中实时读取他们的值是否改变来生成动画。这里以平移模块为例：左图为变换菜单的设置，右图为其中的平移功能在绘制函数 `void gaobin_Triangle(void)` 中的实现。

```

/*变换菜单函数*/
void TransformMenu(int date)
{
    option_value = date;
    switch (option_value)
    {
        case 1://平移
            sumT_Count = 0;
            spin = 0;
            size = 1;
            move = 0.005f;
            break;
        case 2://旋转
            sumR_Count = 0;
            move = 0;
            size = 1;
            spin = 0.5f;
            break;
        case 3://缩放
            sumS_Count = 0;
            move = 0;
            spin = 0;
            size = 1.08f;
            break;
    }
}

/*全局变量*/
static int option_value = 1;           //菜单值
static GLfloat move = 0.0;            //平移量
static GLfloat spin = 0.0;            //旋转量
static GLfloat size = 1.0;            //缩放量
static double sumT_Count = 0;          //平移总距离
static double sumR_Count = 0;          //旋转总度数
static double sumS_Count = 1;          //缩放总数
static double speedlevel = 1;          //速度等级
static double LINE_or_FILL = 0;       //是否填充的判断

//平移模块
if (move != 0)
{
    sumT_Count += move * speedlevel;
    if (sumT_Count <= 1)
    {
        glTranslatef(GLfloat(move * speedlevel), 0, 0);
    }
    if (sumT_Count >= 1 && sumT_Count < 3)
    {
        glTranslatef(GLfloat(-move * speedlevel), 0, 0);
    }
    if (sumT_Count >= 3 && sumT_Count < 4)
    {
        glTranslatef(GLfloat(move * speedlevel), 0, 0);
    }
    if (sumT_Count >= 4)
    {
        move = 0;
    }
}

```

(6) 速度设置上，我们分为三级，低速，中速高速就是分别给变化量乘以一个倍数，这里设置为 1、3、5。

```

/*速度菜单函数*/
void VelocityMenu(int date)
{
    option_value = date;
    switch (option_value)
    {
        case 4://低速
            speedlevel = 1;
            break;
        case 5://中速
            speedlevel = 3;
            break;
        case 6://快速
            speedlevel = 5;
            break;
    }
}

//旋转模块
if (spin != 0)
{
    sumR_Count += spin * speedlevel;
    if (sumR_Count <= 360)
    {
        glRotatef(GLfloat(-spin * speedlevel), 0, 0, 1);
    }
    if (sumR_Count > 360 && sumR_Count <= 720)
    {
        glRotatef(GLfloat(spin * speedlevel), 0, 0, 1);
    }
    if (sumR_Count > 720)
    {
        spin = 0;
    }
}

```

(7) 是否填充，就是直接在绘制前加一个判断。

```

    }
    /*显示菜单函数*/
    void DisplayMenu(int date)
    {
        option_value = date;
        switch (option_value)
        {
            case 7://填充
                LINE_or_FILL = 1;
                break;
            case 8://线框
                LINE_or_FILL = 0;
                break;
        }
    }
}

```

```

/*是否填充*/
if(LINE_or_FILL==0) glPolygonMode(GL_BACK, GL_LINE); //线框模式
else if(LINE_or_FILL==1) glPolygonMode(GL_BACK, GL_FILL); //填充模式

/*三角形*/
glLineWidth(4.0f); // 设置线的宽度
glBegin(GL_TRIANGLES);
    glColor3d(0.0, 1.0, 0.0); //设置颜色
    glVertex2d(0.25*cos(0), 0.25*sin(0));
    glVertex2d(0.25*cos(240 * PI / 180), 0.25*sin(240 * PI / 180));
    glVertex2d(0.25*cos(120 * PI / 180), 0.25*sin(120 * PI / 180));
glEnd();

```