

2018-2019年度第二学期 00106501

计算机图形学



童伟华 管理科研楼1205室

E-mail: tongwh@ustc.edu.cn

中国科学技术大学 数学科学学院

<http://math.ustc.edu.cn/>





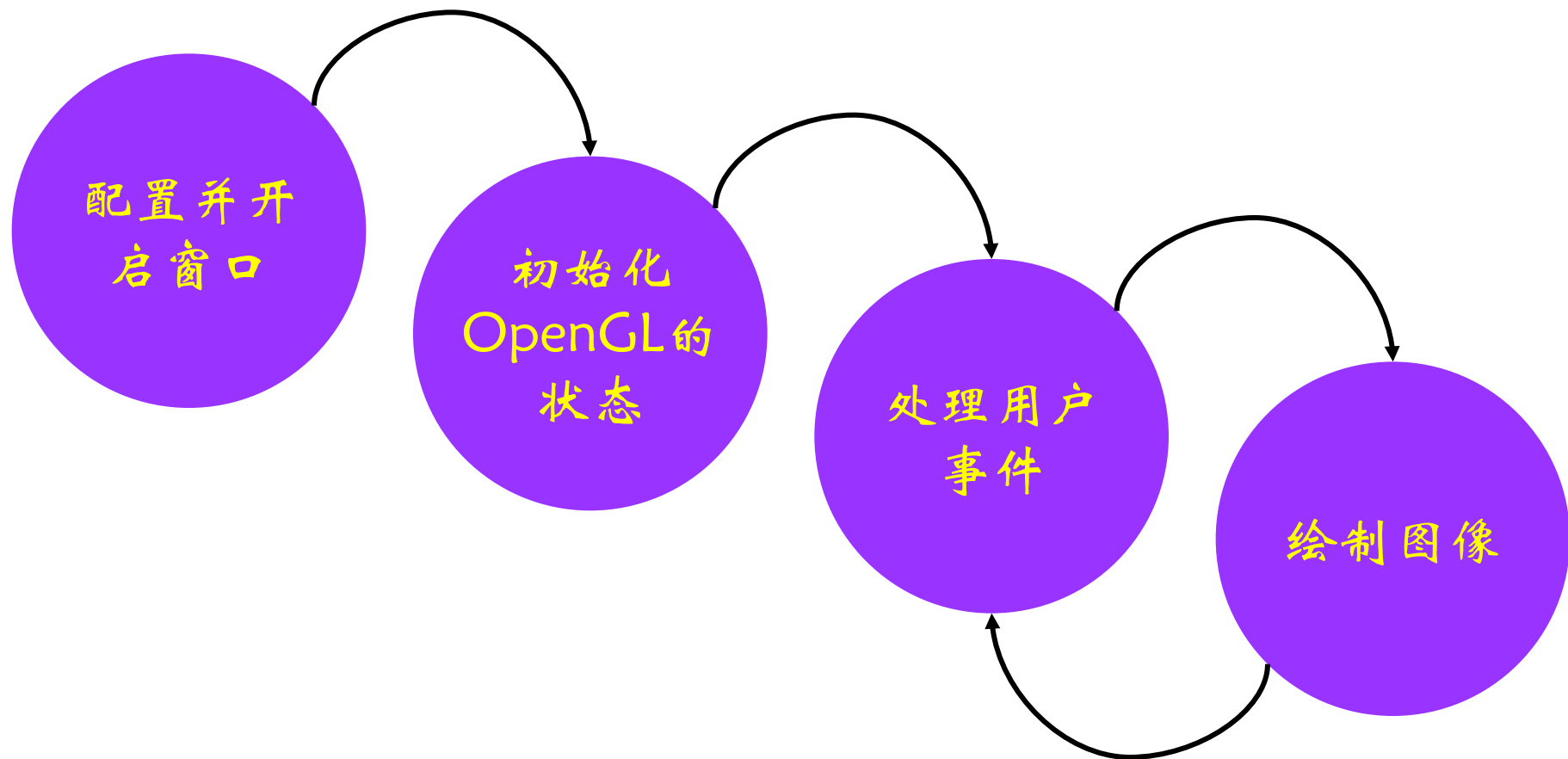
第二节 完整的程序

OpenGL程序的一般结构



- 基于Qt的OpenGL程序具有类似的结构，包含下述函数
 - main():
 - 打开一个或多个具有指定属性的窗口
 - 进入事件循环（最后一条可执行语句）
 - 继承或组合OpenGL相关的类：QOpenGLWidget/QOpenGLWindow，并重载以下函数
 - initializeGL(): 初始化函数
 - paintGL(): 绘制函数
 - resizeGL(): 窗口大小改变函数

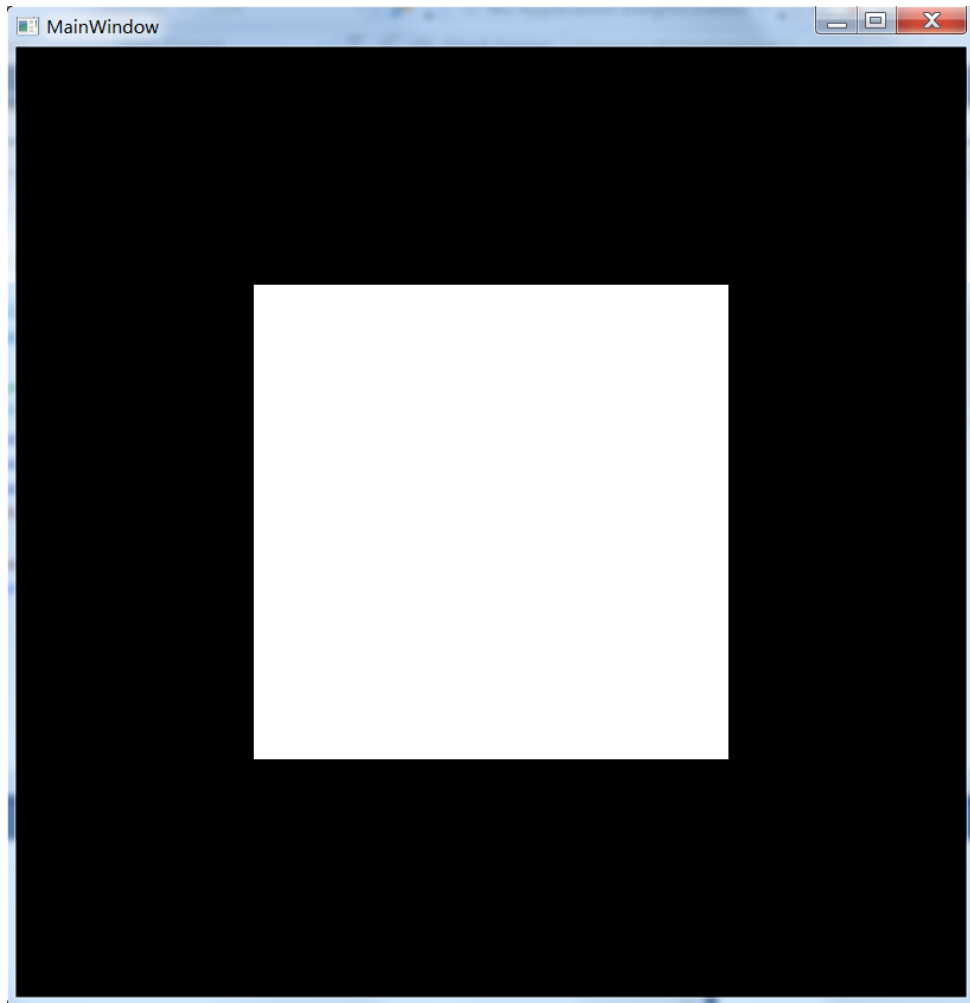
OpenGL程序的一般结构



一个简单程序



- 在黑色背景上画一个白色矩形



main.cpp



```
#include "mainwindow.h"
#include <QtWidgets/QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;

    // setup OpenGL format
    QSurfaceFormat format;
    format.setVersion(4, 6);
    format.setProfile(QSurfaceFormat::CompatibilityProfile);
    format.setSamples(8);
    w.setFormat(format); // must be called before the widget or its
parent window gets shown

    w.show();
    return a.exec();
}
```

mainwindow.h



```
...
class MainWindow : public QOpenGLWidget
{
    Q_OBJECT

public:
    MainWindow(QOpenGLWidget *parent = 0);
    ~MainWindow();

protected:
    void initializeGL() override;
    void paintGL() override;
    void resizeGL(int w, int h) override;

private:
    Ui::MainWindowClass ui;
};
...
```

mainwindow.cpp



```
void MainWindow::initializeGL()
{
    // initialize GLEW library
    glewExperimental = true;
    glewInit();

    // set the background color
    glClearColor(0.2f, 0.3f, 0.3f, 1.0f);
}

void MainWindow::resizeGL(int w, int h)
{
    // set the viewport
    glViewport(0, 0, w, h);
}
```


mainwindow.cpp



```
void MainWindow::paintGL()
{
    // render scenes
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0f, 1.0f, 1.0f);
    glBegin(GL_POLYGON);
        glVertex2d(-0.5, -0.5);
        glVertex2d(-0.5, 0.5);
        glVertex2d(0.5, 0.5);
        glVertex2d(0.5, -0.5);
    glEnd();
    glFlush();
}
```

Qt 中与 OpenGL 相关的类



- 在 Qt 5 中，与 OpenGL 相关的类被分解到 Qt GUI 与 Qt Widgets 模块
 - Qt Widgets 模块：QOpenGLWidget
 - Qt GUI 模块：QOpenGLWindow, OpenGLFunctions, OpenGL Shader, OpenGLProgram, OpenGLBuffer, ...
- 在 Qt 5 中有三种方式使用 OpenGL：
 - 继承 QOpenGLWindow 类，覆盖虚函数：initializeGL(), paintGL(), resizeGL();
 - 继承 QOpenGLWidget 类，覆盖虚函数：initializeGL(), paintGL(), resizeGL();
 - 多重继承 QWindow, QOpenGLFunctions, 通过 setSurface(Qwindow::OpenGLSurface)，实现相关的函数。

覆盖相关的虚函数



初始化OpenGL: initializeGL()

```
void glewInit(void)
```

窗口大小改变回调函数: resizeGL(int w, int h)

```
void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)
```

绘制回调函数: paintGL()

清空缓冲区:

```
void glClear(GLbitfield mask)
```

图元定义: mode可取GL_POINTS、GL_LINES、GL_POLYGON

```
void glBegin(GLenum mode) // 开始mode型对象定义
```

```
void glEnd() // 结束顶点序列
```

强制执行OpenGL命令:

```
void glFlush()
```

注意: 为了降低学习难度, 上面的例子使用compatibility profile编写

事件循环



- 在程序中定义了一个绘制函数（虚函数）：`paintGL()`
 - 只要OpenGL确定显示内容要被刷新时，绘制函数就会被调用：例如，当窗口被打开的时候
 - `main`函数是以程序进入事件循环做为结束

默认值



- 立即绘制模式 (compatibility profile) 非常简单
- 大量使用状态变量的默认值
 - 视图
 - 颜色
 - 窗口参数
- 逐渐过渡到：核心绘制模式 (core profile)

程序里需要什么



■ 头文件

```
#include <GL/glew.h> //自动包含gl.h, glu.h
```

■ 库文件

- 库文件：编译器或系统库文件目录\opengl32.lib glu32.lib glew32.lib
- 动态链接库文件：操作系统目录\system32\opengl32.dll glu32.dll glew32.dll

■ 数据类型

- 为了兼容性，OpenGL定义了各种数据类型(#define)
 - GLfloat, GLint, GLenum, etc.

■ 仅以Visual Studio 2013为例

- 编译GLEW库，并把头文件（glew.h）、库文件（glew32.lib）及动态库文件（glew32.dll）拷贝到相应的目录
- 创建一个Qt GUI Application类型的工程
- 编写C++代码
- 进入菜单 Project → Properties，选择Link标签，Input子项，在Additional Dependencies文本编辑控件中加上opengl32.lib; glu32.lib; glew32.lib（注意用“;”分开）



Thanks for your attention!

