

# Qt 编程基础

中国地质大学（武汉）自动化学院

# 目录

## 1 Qt 简介

- 功能和应用
- 版本和学习资料

## 2 Qt 基础

- 信号和槽函数
- GUI 界面
- 菜单和工具栏
- 控件
- 布局
- 事件
- 模型/视图结构
- 画图

## 3 示例

- 时钟
- 贪吃蛇

## 4 后记

- C++ 程序设计总结
- 推荐书籍
- 编程能力
- 寄语

## 学习目标

- ① 掌握 Qt 开发环境的基本使用
- ② 了解基本的 window 界面框架
- ③ 理解信号与槽机制
- ④ 理解事件机制
- ⑤ 理解模型/视图结构
- ⑥ 学会简单动画绘图

# 1 Qt 简介

跨平台 C++ 图形用户界面应用程序开发框架, <https://www.qt.io/>

- 它既可以开发 GUI 程序，也可用于开发非 GUI 程序，比如控制台工具和服务器。
- Qt 是面向对象的框架，使用元对象编译器生成扩展代码，易于扩展，允许组件编程。

# 1 Qt 简介

跨平台 C++ 图形用户界面应用程序开发框架, <https://www.qt.io/>

- 它既可以开发 GUI 程序, 也可用于开发非 GUI 程序, 比如控制台工具和服务器。
- Qt 是面向对象的框架, 使用元对象编译器生成扩展代码, 易于扩展, 允许组件编程。



## 跨平台

桌面: Windows, Linux, Mac; 移动: iOS, Android, WP; 嵌入式: QNX, VxWorks

# 1 Qt 简介

跨平台 C++ 图形用户界面应用程序开发框架, <https://www.qt.io/>

- 它既可以开发 GUI 程序, 也可用于开发非 GUI 程序, 比如控制台工具和服务器。
- Qt 是面向对象的框架, 使用元对象编译器生成扩展代码, 易于扩展, 允许组件编程。



## 跨平台

桌面: Windows, Linux, Mac; 移动: iOS, Android, WP; 嵌入式: QNX, VxWorks



## 集成开发平台

Design, Code, Debug & Deploy Quickly

# 1 Qt 简介

跨平台 C++ 图形用户界面应用程序开发框架, <https://www.qt.io/>

- 它既可以开发 GUI 程序, 也可用于开发非 GUI 程序, 比如控制台工具和服务器。
- Qt 是面向对象的框架, 使用元对象编译器生成扩展代码, 易于扩展, 允许组件编程。



## 跨平台

桌面: Windows, Linux, Mac; 移动: iOS, Android, WP; 嵌入式: QNX, VxWorks

## 集成开发平台

Design, Code, Debug & Deploy Quickly

应用: WPS、Skype、豆瓣电台、虾米音乐、VirtualBox、Opera、咪咕音乐、SMPlayer、VLC 多媒体播放器、Google 地球、Adobe Photoshop Album、Texmaker、Opera、Qt Creator, Mercedes-Benz,

# 1 Qt 简介

## 软件下载

Qt 版本 Qt 5: <https://www.qt.io/offline-installers>



# 1 Qt 简介

## 软件下载

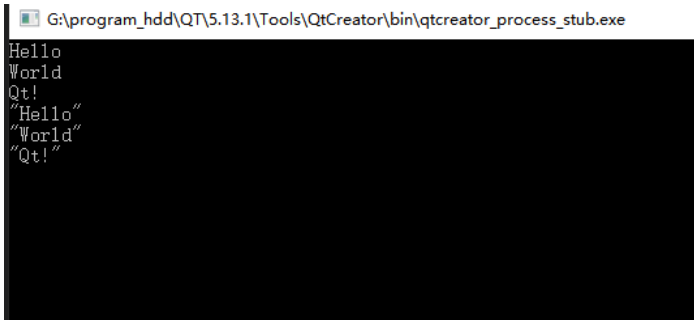
Qt 版本 Qt 5: <https://www.qt.io/offline-installers>

## 在线教程

- Qt 学习之路:
  - <https://www.w3cschool.cn/learnroadqt/>
  - <https://www.devbean.net/2012/08/qt-study-road-2-catelog/>
- Qt Documentation: <http://doc.qt.io/qt-5/gettingstarted.html>
- Qt examples: <http://doc.qt.io/qt-5/examples-widgets.html>

## 2 Qt 基础

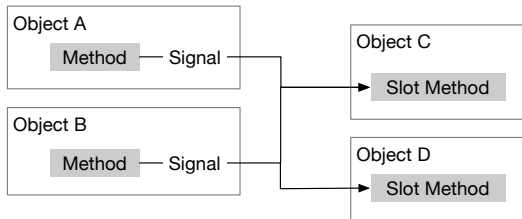
### (1) 创建 Hello World Qt!



```
G:\program_hdd\QT\5.13.1\Tools\QtCreator\bin\qtcreator_process_stub.exe
Hello
World
Qt!
"Hello"
"World"
"Qt!"
```

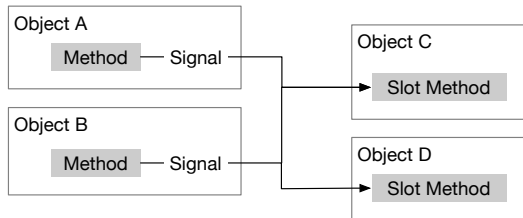
## 2 Qt 基础

### (2) 使用信号与槽机制实现简单的对象间通讯



## 2 Qt 基础

### (2) 使用信号与槽机制实现简单的对象间通讯



#### 信号 signals

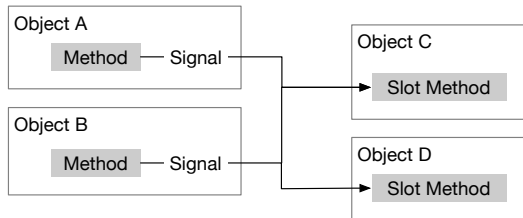
- 信号由对象发射
- 基类的信号被派生类继承
- 无实现，类似 C++ 纯虚函数

#### 槽 slots

- 槽是普通的 C++ 成员函数
- 多个信号可以与其相关联
- 槽可以有参数，但参数不能有缺省值。

## 2 Qt 基础

### (2) 使用信号与槽机制实现简单的对象间通讯



#### 信号 signals

- 信号由对象发射
- 基类的信号被派生类继承
- 无实现，类似 C++ 纯虚函数

#### 槽 slots

- 槽是普通的 C++ 成员函数
- 多个信号可以与其相关联
- 槽可以有参数，但参数不能有缺省值。

信号-槽链接: `connect(sender, signal, receiver, slot)`

**signal** 接口函数声明; **slot** 响应 **signal** 的实现; **sender** 发送消息的对象; **receiver** 接收消息

## 2 Qt 基础

### (2) 使用信号与槽机制实现简单的对象间通讯

#### 示例 1-NewsPaper

G:\program\_hdd\QT\5.13.1\Tools\QtCreator\bin\qtcreator\_process\_stub.exe

Receives Newspaper: "Newspaper A" QDate("2019-10-15")

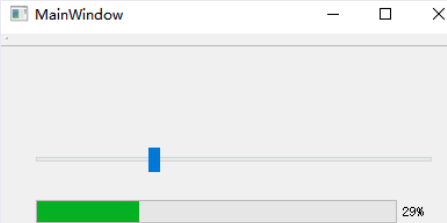
## 2 Qt 基础

### (2) 使用信号与槽机制实现简单的对象间通讯

#### 示例 1-NewsPaper

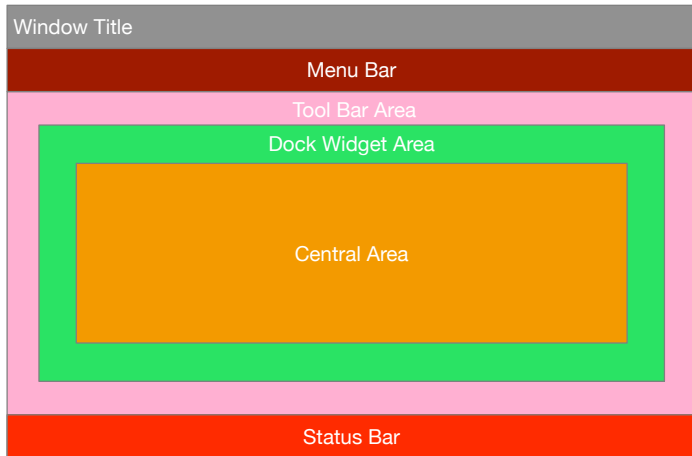
```
G:\program_hdd\QT\5.13.1\Tools\QtCreator\bin\qtcreator_process_stub.exe  
Receives Newspaper: "Newspaper A"   QDate("2019-10-15")  
-
```

#### 示例 2-SliderSignals



## 2 Qt 基础

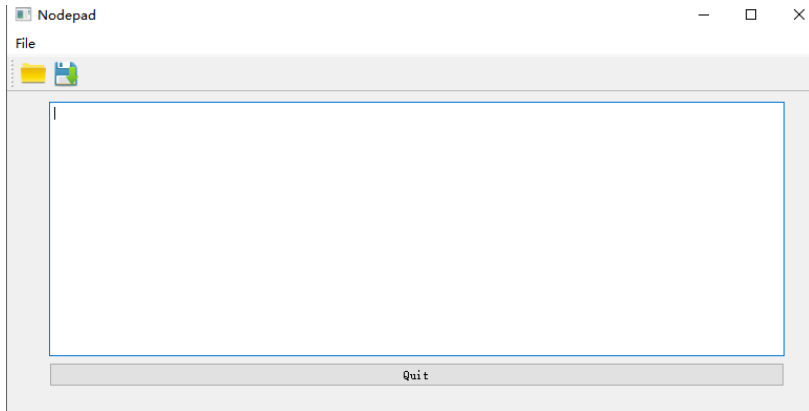
### (3) 创建 GUI 界面





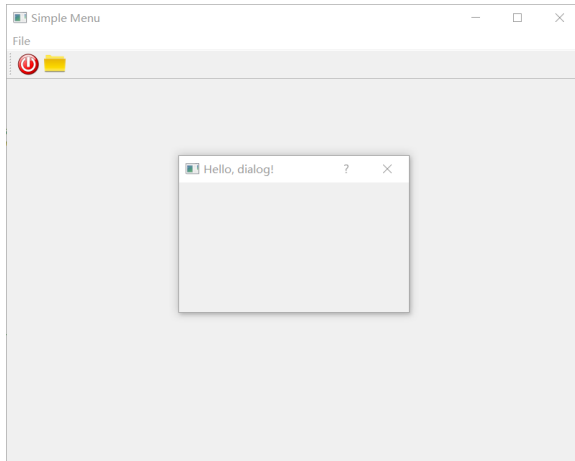
## 2 Qt 基础

### (3) 创建 GUI 界面示例-Notepad



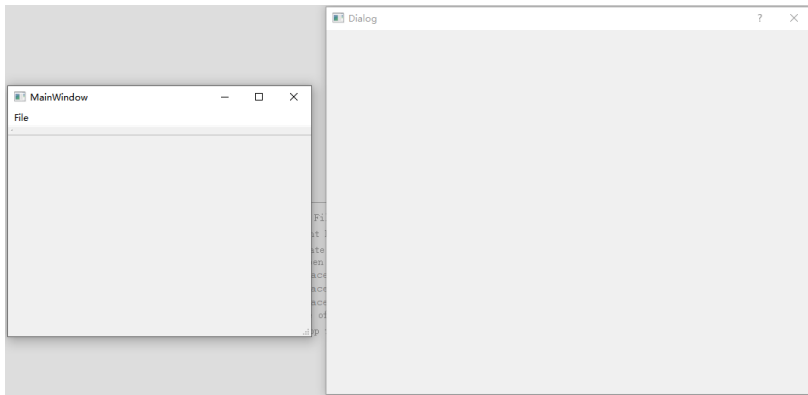
## 2 Qt 基础

### (4) 动作 (action)-示例



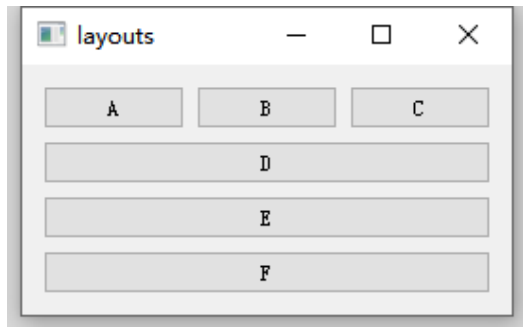
## 2 Qt 基础

### (5) 使用设计模式添加控件示例-dialog



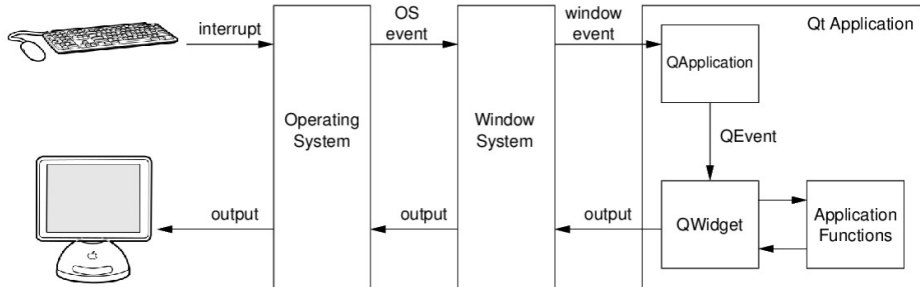
## 2 Qt 基础

### (6) 使用代码编辑 UI 控件布局



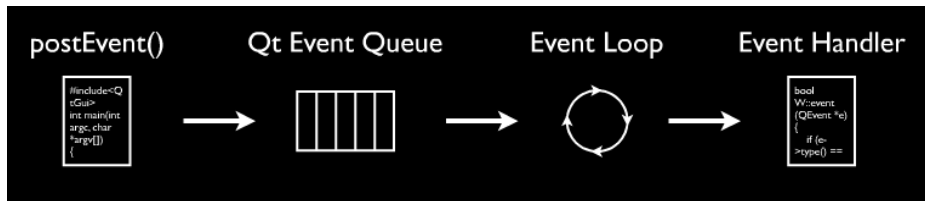
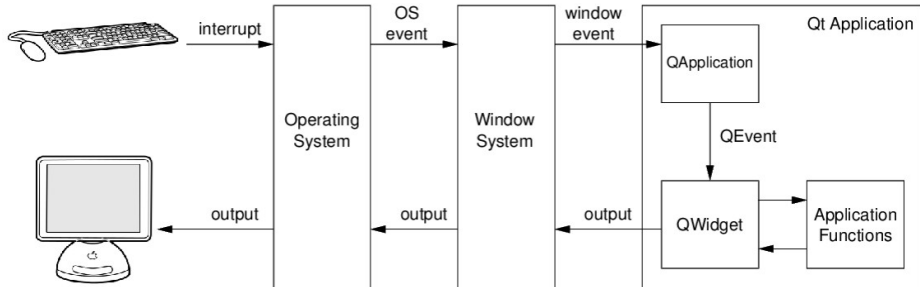
## 2 Qt 基础

### (7) 事件 (event)



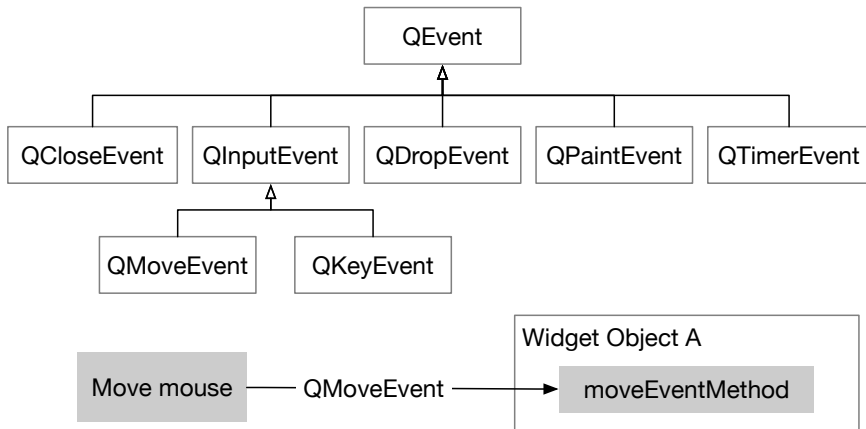
## 2 Qt 基础

### (7) 事件 (event)



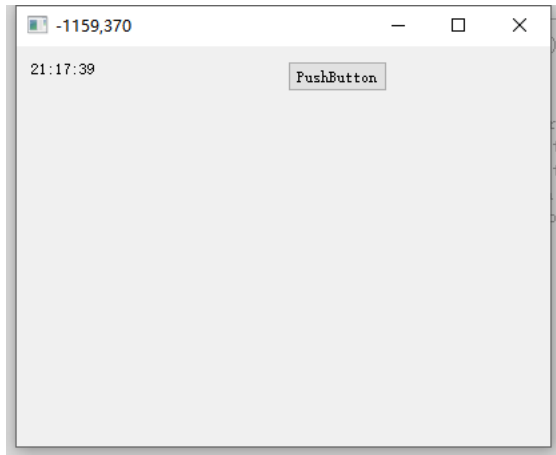
## 2 Qt 基础

### (7) 事件 (event)



## 2 Qt 基础

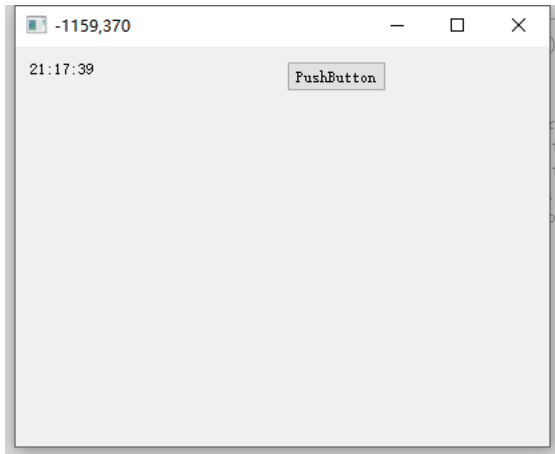
### (7) 事件 (event)-示例





## 2 Qt 基础

### (7) 事件 (event)-示例

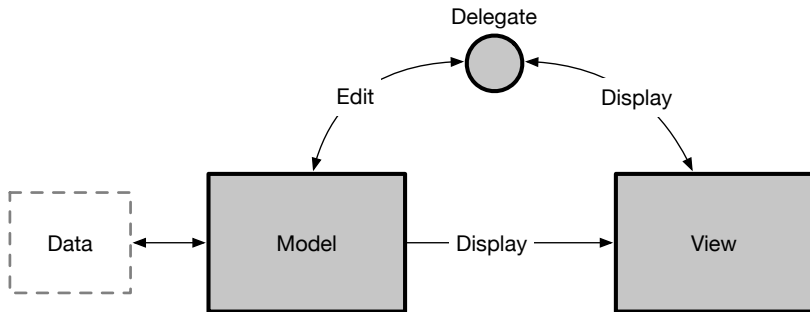


#### 说明

- 事件不能等同于消息和槽机制
- 事件在组件层面上传播，而不是通过继承机制
- 通过事件过滤器接受相应事件或继续转发事件

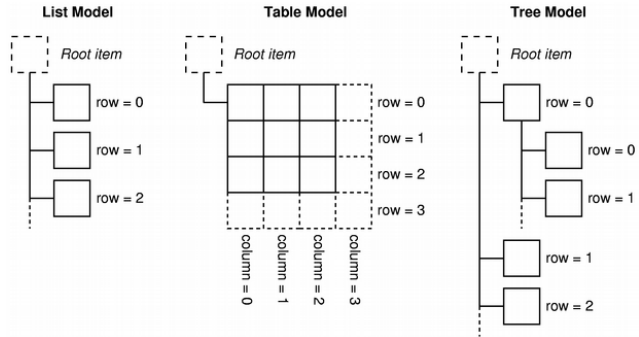
## 2 Qt 基础

### (8) 模型/视图结构 (Model/View)



## 2 Qt 基础

### (8) 模型/视图结构 (Model/View)-示例



## 2 Qt 基础

### (8) 模型/视图结构 (Model/View)-示例

List Model

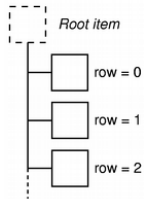
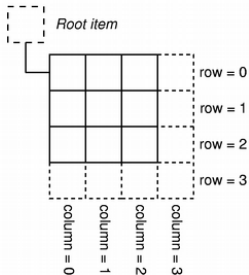
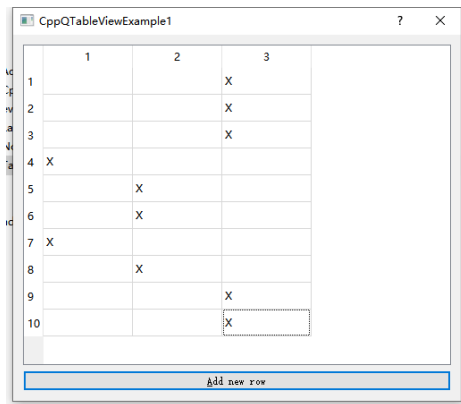
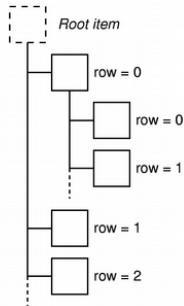


Table Model



Tree Model



## 2 Qt 基础

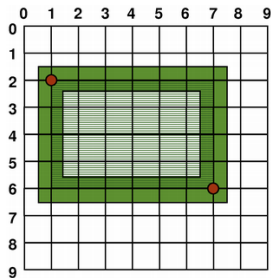
### (9) 画图 (Painting)



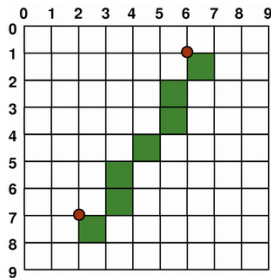
- QPainter (画笔) 用来执行绘制的操作;
- QPaintDevice (画布) 是一个二维空间的抽象, 允许 QPainter 在其上面进行绘制;
- QPaintEngine 提供了 QPainter 在不同的设备上进行绘制的统一的接口。

## 2 Qt 基础

### (9) 画图 (Painting)

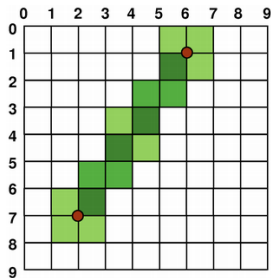


坐标系



```
QPainter painter(this);
```

```
painter.setPen(Qt::darkGreen);  
painter.drawLine(2, 7, 6, 1);
```



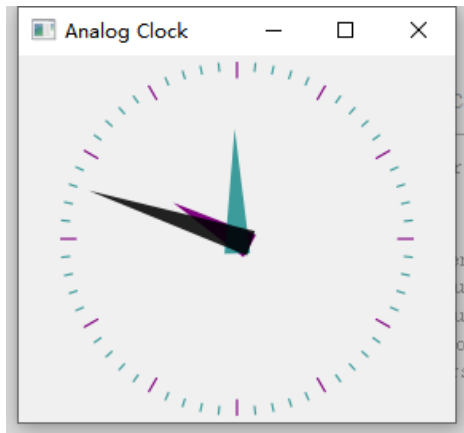
反走样

```
QPainter painter(this);  
painter.setRenderHint(  
    QPainter::Antialiasing);  
painter.setPen(Qt::darkGreen);  
painter.drawLine(2, 7, 6, 1);
```

```
QPainter painter(this);  
painter.setPen(Qt::darkGreen);  
//Using the (x y w h) overload  
painter.drawRect(1, 2, 6, 4);
```

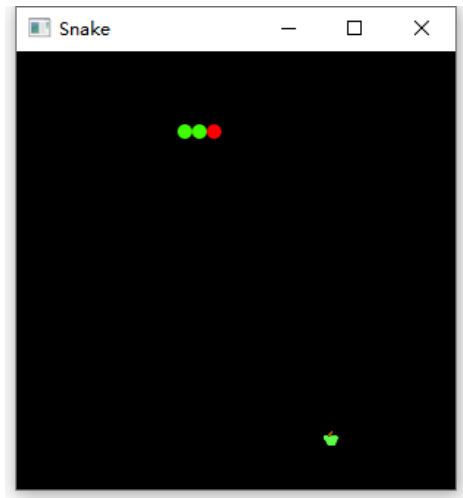
## 3 示例

### 示例一：时钟



### 3 示例

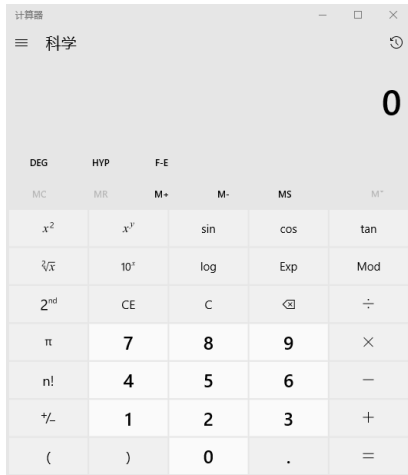
#### 示例二：贪吃蛇





## 第四次验收

仿照 Windows 系统的计算器软件，为教材第 12.4 节通用计算器设计界面，开发一款实用的计算器软件，并按照实验指导书附录中课程设计报告模板要求撰写结课报告。



## 结课考核

学生成绩管理系统，按照实验指导书附录中课程设计报告模板要求撰写结课报告。

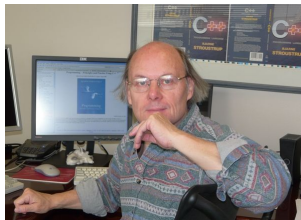
## 4 后记

### C++ 编程风格

- 数据抽象
- 过程化
- 面向对象
- 泛型编程

### C++ 知识层级

- 层级一：语法/语意
- 层级二：专家经验
- 层级三：底层机制
- 层级四：设计观念复用



### 网友的评价

- **精通 C++ 是一个艰巨的任务。**为什么 C++ 比别的语言难学这么多？是因为 C++ 他爹 Bjarne Stroustrup 说过的一句话“我特别讨厌语言的设计者把自己的喜好强加给用户” C++ 能够自由的让你放弃某些部分，而别的语言会阻止你放弃某些部分。
- 谷歌工程师对 C++ 的掌握有两个级别：拥有 C++ 的 readability（可读性）认证；顾问级 C++ 程序猿
- Never trust a programmer who says he knows C++

## 4 后记

### 推荐 C++ 书籍

- 层级一：语法/语意 (C++)

- ① C++ Primer ( 中文版, 侯俊杰译) by Stanley B. Lippman

- 层级二：专家经验 (C++/OOP)

- ① (More )Effective C++(中文版, 侯俊杰译), by Scott Meyers.
- ② (More )Exceptional C++ (中文版, 侯俊杰译), by Herb Sutter
- ③ Effective Modern C++, by Scott Meyers

- 层级三：底层机制 (C++ Object Model)

- ① Inside the C++ Object Model (深度探索 C++ 物件模型, 侯俊杰译),by Stanley Lippman.

- 层级四：设计观念的复用 (C++/Patterns)

- ① Design Patterns: Elements of Reusable Object Oriented Software, by Erich Gamma,Richard Helm,Ralph Johnson,and John Vlissides
- ② Modern C++ Design: Generic Programming and Design Patterns Applied by Andrei Alexandrescu.

## 4 后记

编程能力:运用机器解决问题的能力

- 深度：算法，数据结构，强化学习，模式识别、深度学习、智能优化
- 广度：操作系统，分布式系统，存储系统，游戏引擎、数据库、GPU, FPGA, AR/VR

AI 之路漫漫其修远兮，吾将上下而求索