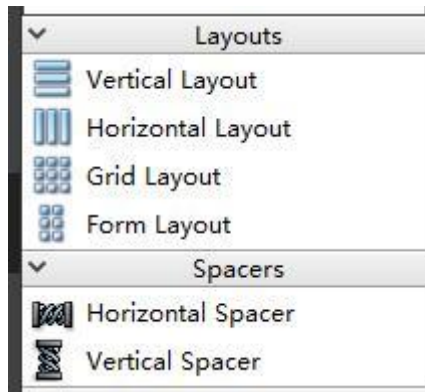


## 2010--开发常用控件(Layouts & Spacers)--零声教育 vico 老师

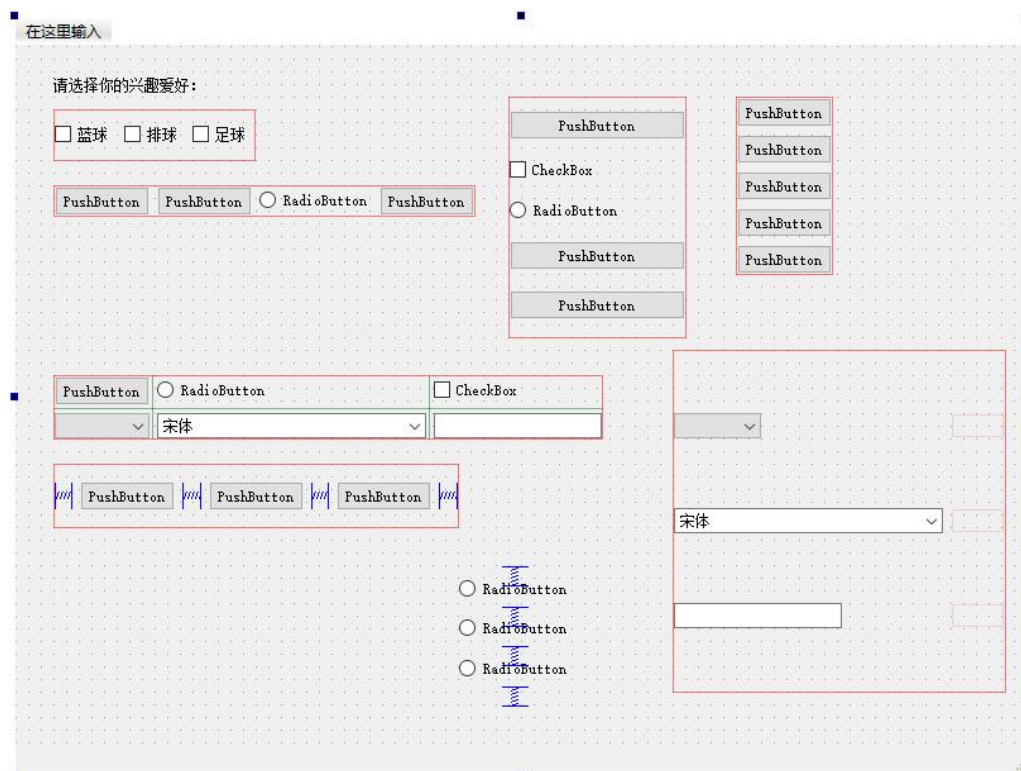
## 【Qt5 布局管理和空间间隔】



**控件名称依次解释如下:**

- Vertical Layout: 垂直布局。
- Horizontal Layout: 横向（水平）布局。
- Grid Layout: 网格（栅格）布局。
- Form Layout: 表单布局。
- Horizontal Spacer: 水平间隔。
- Vertical Spacer: 垂直间隔。

### 手动操作:



**【全部代码案例分析，大家直接看工程即可】**

**1、GridLayout:Grid Layout 网格布局（栅格布局）：**此控件划分行和列组成，  
能够将控件放入一个网状栅格当中，每一个控件占一格或多行多列。

**头文件：**

```
#ifndef WIDGET_H
#define WIDGET_H

#include <QWidget>

#include <QGridLayout> // 网格控件头文件
#include <QLabel> // 标签控件头文件
#include <QPushButton> // 命令按钮控件头文件

class Widget : public QWidget
{
    Q_OBJECT

public:
    Widget(QWidget *parent = nullptr);
    ~Widget();

    QGridLayout *pGrid_layouts;

    QPushButton *button1;
    QPushButton *button2;
    QPushButton *button3;
    QPushButton *button4;

};
#endif // WIDGET_H
```

**源码文件：**

```
#include "widget.h"

Widget::Widget(QWidget *parent)
    : QWidget(parent)
{
```

```

        button1=new QPushButton(this);
        button1->setText("第一区：顶部菜单栏选项");
        button1->setFixedHeight(40); // 设置固定大小高度

button1->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::Expanding)
;

        button2=new QPushButton(this);
        button2->setText("第二区：侧边栏选项");
        button2->setFixedWidth(100); // 设置固定大小宽度

button2->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::Expanding)
;

        button3=new QPushButton(this);
        button3->setText("第三区：底部选项");
        button3->setFixedHeight(40);

button3->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::Expanding)
;

        button4=new QPushButton(this);
        button4->setText("第四区：子窗体选项");

button3->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::Expanding)
;

        pGrid_layouts=new QGridLayout();

// 通过此函数设置左侧 顶部 右侧 底部边距, 主要方便布局周围进行使用
pGrid_layouts->setContentsMargins(0, 0, 0, 0);

// pGrid_layouts->setMargin(30);
// pGrid_layouts->setSpacing(40);
pGrid_layouts->setSpacing(0);

// 显示位置
// addWidget(参数 1, 参数 2, 参数 3, 参数 4, 参数 5, 参数 6)
/*
1:我要插入的子布局对象
2:插入的开始行
3:插入的开始列
4:占用的行数

```

```

5:占用的列数
6:指定对齐方式
*/
pGrid_layouts->addWidget(button1, 0, 1);
pGrid_layouts->addWidget(button2, 0, 0, 3, 1);
pGrid_layouts->addWidget(button3, 2, 1);
pGrid_layouts->addWidget(button4, 1, 1);

setLayout(pGrid_layouts);

}

Widget::~Widget()
{

}

```

## 2、表单布局

头文件:

```

#ifndef WIDGET_H
#define WIDGET_H

#include <QWidget>

class Widget : public QWidget
{
    Q_OBJECT

public:
    Widget(QWidget *parent = nullptr);
    ~Widget();
};
#endif // WIDGET_H

```

源文件:

```

#include "widget.h"

#include <QFormLayout>
#include <QLineEdit>

```

```

Widget::Widget(QWidget *parent)
    : QWidget(parent)
{

    setFixedSize(250, 200);

    // 创建表单布局指针
    QFormLayout *qLayout=new QFormLayout(this);

    QLineEdit *le1=new QLineEdit(); // 输入学号
    QLineEdit *le2=new QLineEdit(); // 输入姓名
    QLineEdit *le3=new QLineEdit(); // 输入学校

    qLayout->addRow("学号", le1);
    qLayout->addRow("姓名", le2);
    qLayout->addRow("学校", le3);
    qLayout->setSpacing(8);

    // WrapAllRows 将标签显示在单行编辑框上面
    // qLayout->setRowWrapPolicy(QFormLayout::WrapAllRows);

    // 当标签和单选编辑框, 将标签显示在同一行。
    qLayout->setRowWrapPolicy(QFormLayout::WrapLongRows);

    qLayout->setLabelAlignment(Qt::AlignLeft); // 设置标签对齐方式

    setWindowTitle("表单布局测试案例");

}

Widget::~~Widget()
{
}

```