

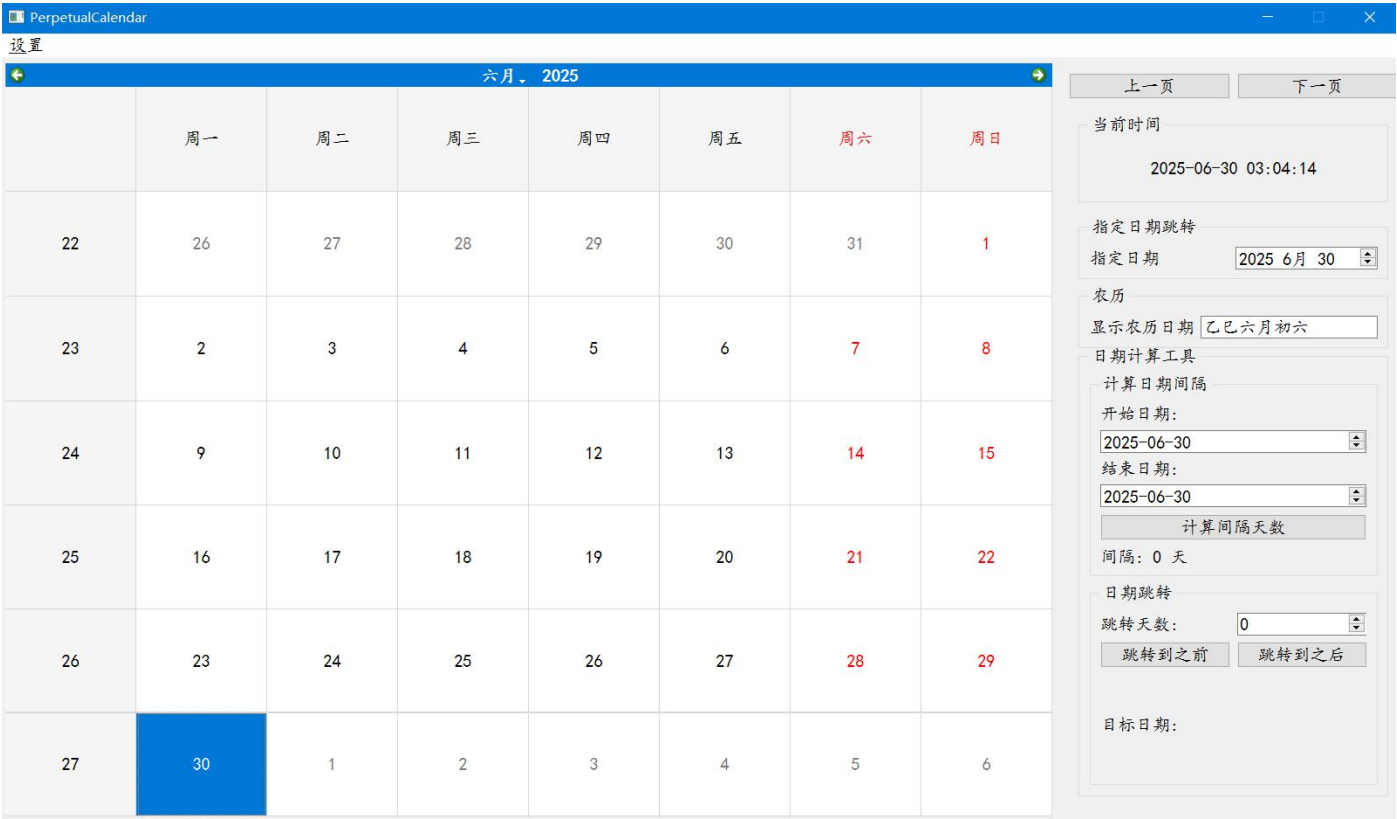
我们组使用 Qt 制作了一份日历，能够显示 1900-2100 年间的日期，并实现一些功能，以下是作业报告的细节。

(1) 程序功能介绍：

本项目实现了以下功能：

- 完整的日历界面；
- 左上角设置可以选择切换网格显示与颜色；
- 显示当地时间（与电脑时间一致）；
- 跳转到指定日期，在框中输入日期即可跳转；
- 农历日期显示，会显示当前日期对应的农历日期；
- 日期计算工具，分为两部分：第一部分是计算两个日期之间的间隔，第二部分是在当前选定日期的基础上跳转至指定天数前/后，并在下方显示目标日期。

以下是项目运行截图，包含以上所有功能。

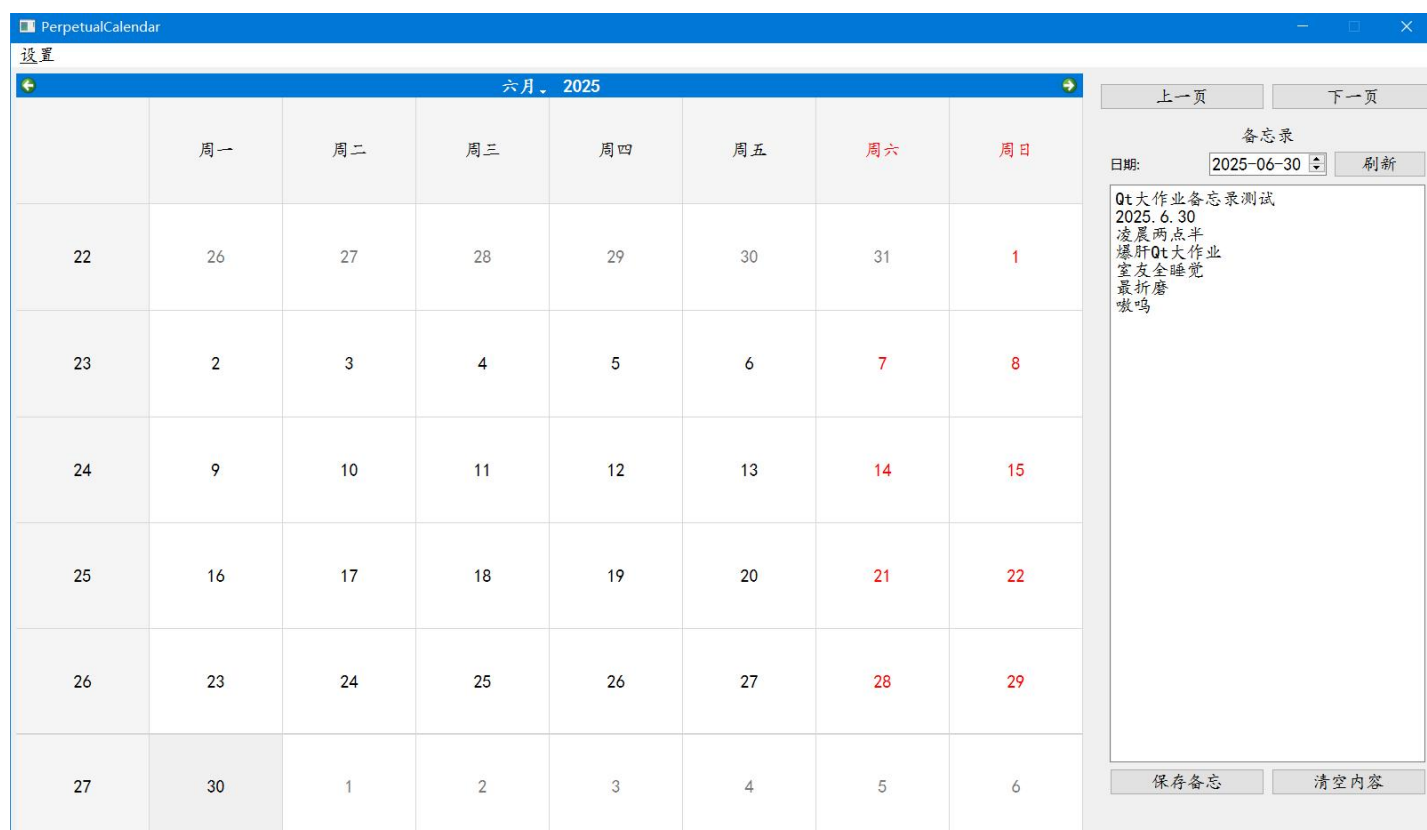


(接下页)

- 因为右侧侧边栏排版原因，备忘录功能实现在了侧边栏第二页（因此也顺带实现了侧边栏翻页功能），具体如下：

既可以在输入日期后，点击“刷新”按钮显示当日的备忘录并且跳转到指定日期（调用了日期跳转函数）；也可以在左侧日历界面点击对应日期并显示当日备忘。

输入备忘后点击下方“保存”可以将备忘储存在 `build` 目录下的 `/CalendarMemos` 文件夹中，点击“清除”可以快速清除当日备忘。



- 除此之外，可以在代码中修改 `getAppFont()` 函数以实现整个界面的字体、大小参数。

（接下页）

(2) 项目各模块与类设计细节:

首先是各功能的模块设计:

- 主窗口模块 (MainWindow)

作为界面主窗口，协调所有子模块，函数功能有如下:

初始化应用界面大小和标题并采用网格布局;使用工厂方法模式创建各子组件;使用堆栈窗口实现侧边栏翻页;实现字体统一管理。

(附源码以及注释)

```
MainWindow::MainWindow(QWidget *parent)
: QMainWindow(parent)
{
    const int width(1280);
    const int height(720); //主界面大小, 宽高
    setWindowTitle(tr("PerpetualCalendar"));
    setFixedSize(width, height); //固定窗口大小

    // Calling Creates
    createMenu();
    createCalendar();
    createDateSelector(); //日期选择
    createNL(); //农历日期
    createTimeDisplay(); //当前时间
    createDateCalculator(); //计算日期以及多少天跳转

    // 初始化备忘录目录
    memoDir = QDir::currentPath() + "/CalendarMemos"; //当前项目的build文件下
    QDir dir;
    if (!dir.exists(memoDir)) {
        dir.mkpath(memoDir);
    }

    //创建侧边栏
    QWidget *cebian = new QWidget();
    sidebar = new QVBoxLayout;

    //创建翻页按钮
    QHBoxLayout *pageButtonsLayout = new QHBoxLayout;
    prevPageButton = new QPushButton("上一页");
    prevPageButton->setFont(getAppFont());
    nextPageButton = new QPushButton("下一页");
    nextPageButton->setFont(getAppFont());
    pageButtonsLayout->addWidget(prevPageButton);
    pageButtonsLayout->addWidget(nextPageButton); //上下页按钮以及字体设置

    //创建堆栈窗口用于翻页
    stackedWidget = new QStackedWidget;
    stackedWidget->addWidget(createPage1()); // 第一页
    stackedWidget->addWidget(createPage2()); // 第二页 (备忘录)

    //将翻页按钮和堆栈窗口添加到侧边栏布局
    sidebar->addLayout(pageButtonsLayout);
    sidebar->addWidget(stackedWidget);

    //侧边栏布局
    cebian->setLayout(sidebar);
    cebian->setFixedWidth(width * 0.25);
    cebian->setFixedHeight(height - menubar->height());

    //创建主布局 (网格布局)
    layout = new QGridLayout;
    layout->addWidget(calendar, 0, 0);
    layout->addWidget(cebian, 0, 1);

    layout->setRowMinimumHeight(0, height - menubar->height());
    layout->setColumnMinimumWidth(0, calendar->width());
    layout->setSizeConstraint(QLayout::SetFixedSize);

    //设置中央部件
    QWidget *central = new QWidget(this);
    central->setLayout(layout);
    setCentralWidget(central);

    //设置边距
    centralWidget()->layout()->setContentsMargins(5, 5, 5, 5);

    //连接翻页按钮信号到槽函数
    connect(prevPageButton, &QPushButton::clicked, this, &MainWindow::prevPage);
    connect(nextPageButton, &QPushButton::clicked, this, &MainWindow::nextPage);
}
```

- 日历模块 (Calendar)

大部分是 Qt 自带 Calendar 项目中的。

- 日期选择模块 (Date Selector)

核心组件: QDateEdit 日期选择器; QGroupBox 分组容器。

实现如下功能: 提供精确日期跳转; 输入框与日历组件双向同步

- 农历显示模块 (Lunar Calendar)
核心组件: QLineEdit 显示农历日期; JSON 数据文件存储农历信息。
实现如下功能: 使用 json 文件存储并加载显示当前日期的农历信息, 并且自动随日期变化更新。
- 时间显示模块 (Time Display)
核心组件: QLabel 时间显示标签; QTimer 定时刷新。
实现如下功能: 显示当前日期和时间并且每秒自动更新
- 日期计算模块 (Date Calculator)
核心组件: QDateEdit 日期输入; QSpinBox 天数输入; QPushButton 计算/跳转按钮。
实现以下功能: 两个日期之间的间隔计算并且自动处理日期顺序(开始日期不能晚于结束日期); 支持当前选定日期±10年范围的日期跳转。
- 备忘录模块 (Memo)
核心组件: QTextEdit 备忘录编辑框; QDateEdit 备忘录日期选择; QLabel 状态显示。
实现如下功能: 自动在 build 文件夹下创建备忘录目录; 按日期以 txt 格式保存/加载备忘录; 支持内容清空; 根据输入、保存、加载等等进行状态反馈, 并且状态消息会自动清除(2-3秒); 与日历组件双向同步。
- 侧边栏翻页系统 (Sidebar Paging)
核心组件: QStackedWidget 堆栈窗口; QPushButton 翻页按钮。
实现如下功能: 两个页面的切换(主功能页/备忘录页)。

类设计细节: 整体采用 MainWindow 类, 其成员变量如下:

公有成员: 构造函数与析构函数。

```
public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();
```

私有成员(信号槽): 实现各类交互, 具体见注释。

```
private slots:
    // Date Selector日期选择
    void selectedDateChanged();

    // Menu主界面
    void setWERed();
    void setWEBlu();
    void setWEBlk();
    void setWECclr(QColor col);

    void setWDRed();
    void setWDBlu();
    void setWDBlk();
    void setWDCclr(QColor col);

    void toggleGridMode();

    void prevPage();
    void nextPage(); // 翻页
    void saveMemo();
    void clearMemo(); // 备忘
    void loadMemoForDate();
```

私有成员：各模块函数以及对应的函数中需要用到的变量等等，以下截取一部分源码。

```
private:
    // Menu
    void createMenu();
    QMenuBar *menubar;
    QMenu *settingsMenu;
    QWidget* createPage1();
    QWidget* createPage2();//为了翻页

    // Calendar
    void createCalendar();
    QCalendarWidget *calendar;
    QGridLayout *layout;
    QVBoxLayout *sidebar;

    // Date Selector
    void createDateSelector();

    QGroupBox *dateSelect_box;
    QDateEdit *dateSelect;
    QLabel *dateSelect_label;

    //农历日期
    void createNL();
    void NL();
    QGroupBox *NL_box;
    QLabel *NL_label;
    QLineEdit *NL_data;

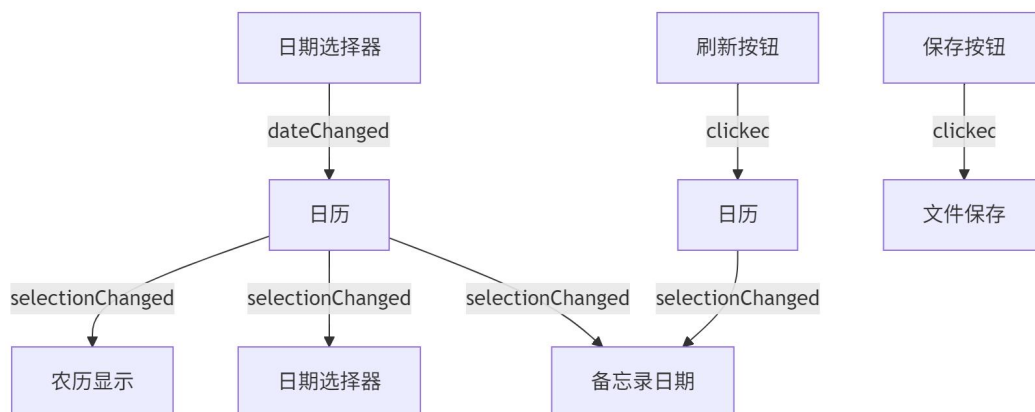
    //系统时间
    QLabel *timeLabel;
    QTimer *timer;
    QGroupBox *timeDisplay_box;
    void updateTimeDisplay();
    void createTimeDisplay();
```

在所有类的方法中，较为关键的是：

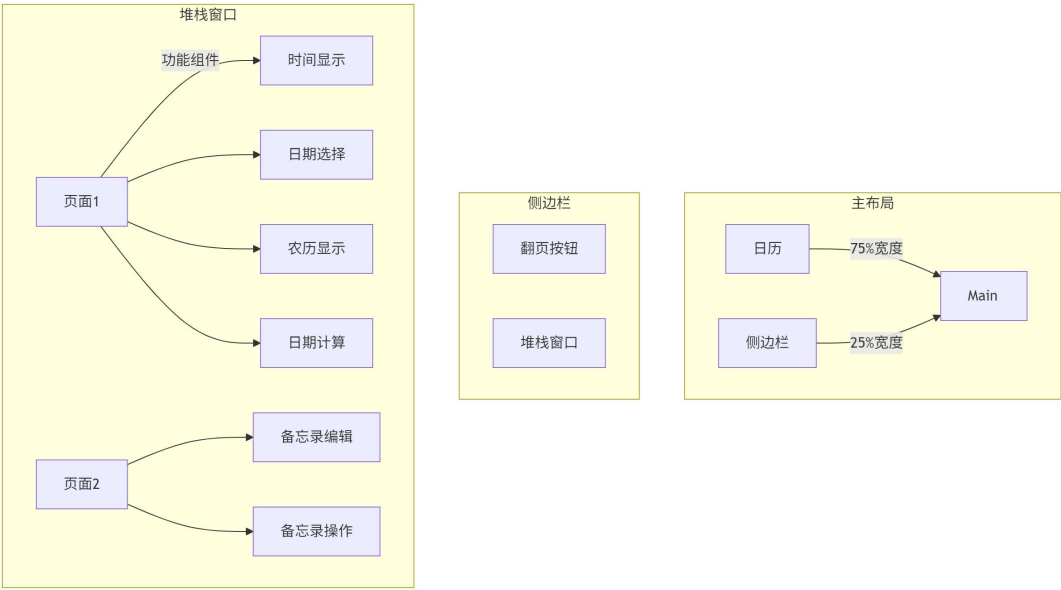
- createXxx(): 组件创建工厂方法
- getAppFont(): 统一字体管理
- createPage1()/createPage2(): 页面构建
- 事件处理槽函数（prevPage, saveMemo 等）

以下是整体的信号槽通信设计：

即，日期选择器与日历双向反馈；而备忘录页面通过刷新按钮与日历进行绑定，所以无论是在左侧点击日期还是在右侧点击刷新都可以同步到另一侧进而加载对应的备忘录内容。



以下是整体的 UI 布局设计：



以下是数据存储与读取：

农历日期以 json 文件存储，并且通过自带的“资源”功能读取并加载。



备忘录以 txt 文件存储在编译产生的/build……/CalendarMemos 目录下，如果本地测试，直接在测试完成后打包删除/build……目录以及项目目录即可，不会赖在电脑的某一个角落。

(3) 小组成员与分工情况：

2100013090 陈文健 负责所有的大作业代码（是的我的组员一点没动），以及此作业报告的撰写。

2400013096 邢一博 负责 5 月 31 日大作业阶段进展材料。

2400012912 李思祺 一起讨论项目功能设计。

(4) 项目总结与反思：

Qt 好好玩，但是下次不想玩了。