

基于 Qt 的时间管理兼自习监督软件设计

8 号, 选择困难症犯了队; 组长: 张子诺; 组员: 倪嘉怡, 杨斯淇

一、程序功能介绍

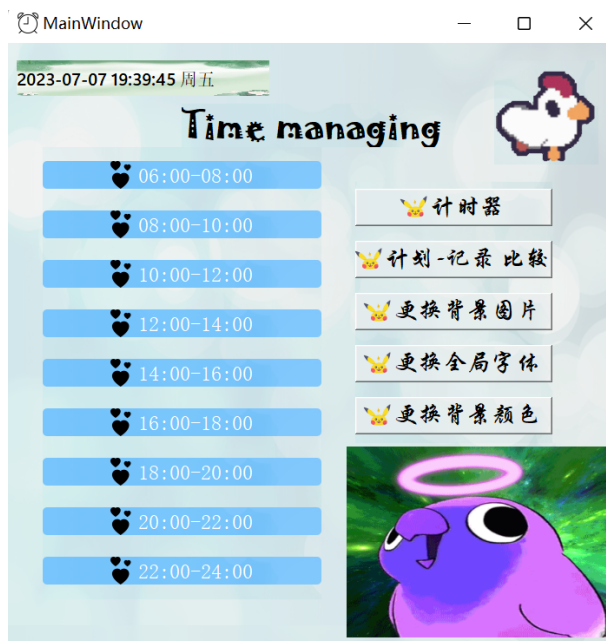


图 1 主界面

“Time-managing” 小程序, 旨在进行时间管理, 实现的各种功能有:

1.1 对一天内的各个时间段进行计划与记录

1. 将一天的时间分划为九个时间段, 帮助用户养成区块化、规律化、健康化的生活习惯与作息习惯。
2. 可以填写各个时间段的计划与记录的名称与内容, 如果用户没有填写, 默认显示 nothing。
3. 对每一次填写的计划与记录内容, 用户可以自己选择是否保存或者清除。对于填写字数过多或计划/记录名称为空的情况进行警告, 防止错误产生。
4. 对于每一项输入的计划, 可以设置是否需要时间提醒。我们的程序将在该任务前一小时内提醒, 提醒持续十秒钟, 以信息框的形式出现在电脑右下角, 简约清晰。
5. 对于计划, 记录内容, 是否提醒, 都可以实现存档功能。程序关闭不影响保存, 支持多次打开和多次修改。



图 2 plan and record

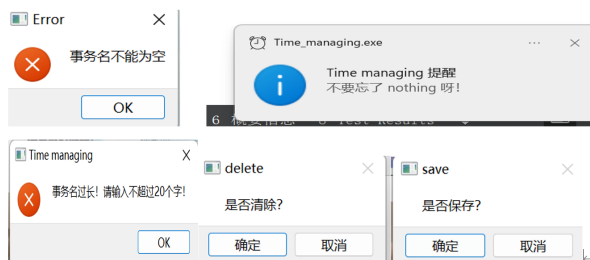


图 3 提示

1.2 计时功能

1. 可以设置计时任务的名称。
2. 用户自由选择时间长短，包括 30，60，90，120,150,180 分钟等不同跨度选择，科学规划专注时间。
3. 计时过程中可以选择无声模式（静音），或 mp3/wav 格式的 BGM，循环播放，协助专注。本程序支持任何用户自己存储的 mp3/wav 格式的音乐。(BGM 可以覆盖主界面)



图 4 计时器

1.3 对计划与记录做可视化对比

为了实现“管理与成长”的初衷,实现可视化,使得用户对时间利用,计划实现有直观的感受。

1. 为每一个时间段建立了直观清晰的已输入计划-记录对比,帮助更好地督促自己完成任务;强调记录的部分,也是希望减轻焦虑,便于脚踏实地往前走。只读,不可修改或增添。
2. 特别设置时间利用率与感想反思两个可供用户在回顾时修改的部分,便于从时间利用率与学习收获两个角度自我评价。
3. 针对每一个区块的这两部分都有存档功能,点击“save”保存,最多可以存储一天九个区块的内容。关闭页面再打开也可以继续编辑或查看。这一部分的档案保存在txt文件中,方便用户自行取用和存储。
4. 可视化部分,设置直观的时间轴和趣味图标,点击有图标放大特效突出选择的时间段对象。



图 5 对比图

1.4 计划列表与记录列表

设计“一日计划一览”与“一日实操记录”两个页面，实现了对之前输入的任务与记录的统筹性可视化。点击“更新”按钮，用户可以直观看到按照时间轴排列的计划/记录总览，便于安排。



图 6 计划



图 7 记录

1.5 程序关闭后最小化到托盘

可以在托盘处选择关闭程序或者重新打开。

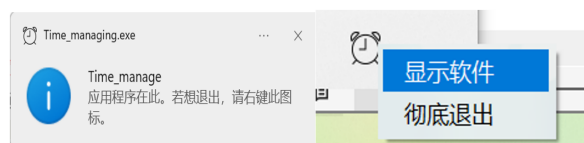


图 8 托盘

1.6 更换背景图片

1. 支持用户选择背景图片, 主界面, 计划记录对比界面, 任务记录界面会随之改变背景。
2. 图片可以从用户电脑自带的图片源选取, 支持 *png/jpeg/jpg* 格式。用户可以按照自己心意打造自己的个性化时间管理软件。

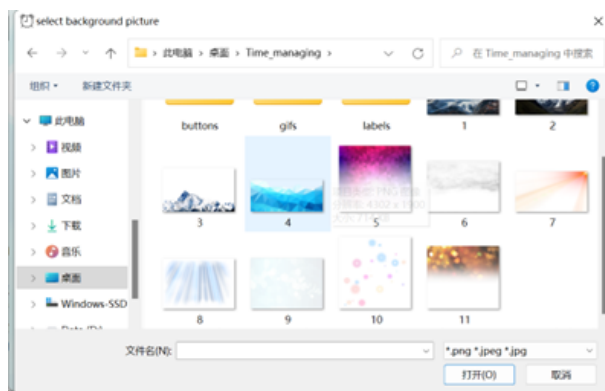


图 9 选择背景图片

1.7 更换背景颜色

1. 如果用户更喜欢简约的风格, 可以选择更换背景颜色, 计划记录对比界面, 任务记录界面会设置为该颜色。
2. 支持选择 Basic colors, 支持在颜色光谱上任意选择自己构造的颜色。
3. 支持用户自己设定 RGB 值, 支持用户通过 html code 设置颜色 (HTML 与不同颜色一一对应), 支持用户调整色调, 明度和饱和度的数值。
4. 支持用户自己添加喜欢的颜色至 custom colors, 定制记忆性个性化审美方案, 方便下次选择。



图 10 更换背景颜色

1.8 更换全局字体

1. 可以选择 Qt 拥有的全部字体, 字号, 特殊效果 (粗体, 删除线, 下划线), 主界面字体随之更改。

2. 点击“确定”前有字体预览功能。

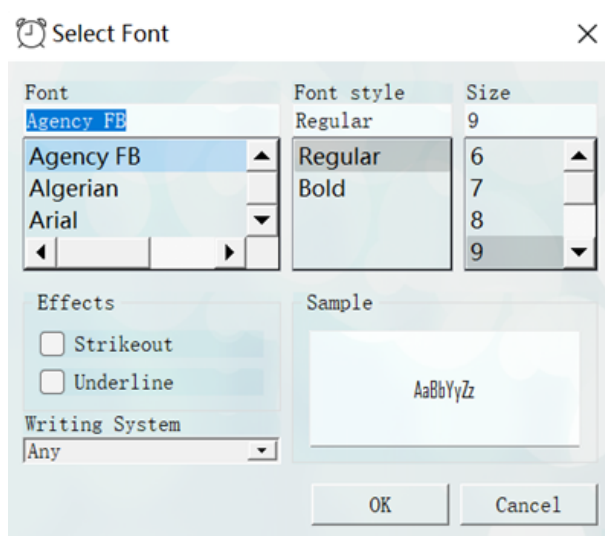


图 11 设置字体

1.9 其他

1. 各界面显示实时时间。
2. ui 进行美化, 主界面设置动图, 按钮加入背景图片/图标。

二、项目各模块与类设计细节

我们在 main.cpp 规定了界面的大小。

一共存在的主要类有：MainWindow 类,task 类,Alltasks 类,task_edit 类,Timerecorder 类,Compare 类。

2.1 MainWindow(主界面)

1. 初始化成员变量, 设置初始 UI, 设置托盘, 设置 Gif, 设置时间显示;
2. 手动/自动关联各个 QPushButton 和对应的槽函数;
3. 对每一个对应的槽函数进行内部逻辑编写。
time_clicked,time_updated,remind_time,changeBackgroundButton,
changeColorAction,fontButton,on_timerecord_clicked,closeEvent,exit_all. 包括弹出窗口、读取路径、更换 UI 等的实现。
4. 关闭程序前进行存档。

2.2 task 类

明确任务的成员变量, 重载成员函数。

(name,content,timenum,is_finish,need_remind,reflection,timeusage)

2.3 Alltasks 类

存储任务, 含多个 task 类。(Allmyplan[10],Allmyfinish[10]) 利用 txt 文档存储信息, 包括所有计划和记录以及是否需要提醒, cpp 文件中实现写入、读入文档功能。

2.4 task_edit 类

1. 设置初始 UI, 加入 Gif 动图, 读取 Alltasks 存储。
2. 关联槽函数, 设置 save(on_complete_clicked 函数) 和 delete_all(on_sweep_clicked 函数) 两个按钮的细节 (利用自定义和 critical 的 QMessageBox), 设置 time_update 函数。
3. 实现存档函数, 在构造函数与析构函数中读写文档。

2.5 Timerecorder 类

1. 设置初始 UI, 加工 Gif 动图。
2. 关联 button 与槽函数。
包括时间显示的 time_updated 函数;
设置好时间 (QComboBox 中) 后的 on_start_clicked,secsToTime,daojishi 函数;
选择与播放音频的
onMediaStatusChanged,on_selectAudio_clicked,on_playAudio_clicked 函数。

2.6 Compare 类

1. 设置初始 ui, 加入按钮及背景图, 创建自己的 allmyplan1[10] 与 allmyfinish1[10] 的任务指针数组, 读取 Alltasks 存储。
2. 运用 stackedWidget 建立三个页面及与之对应的按钮
3. 实现 pressed、released 两个槽函数, 并推广到九个按钮。九个槽函数各自传参连接到 timeblock_clicked 函数, 实现界面显示。
4. 实现存档函数 on_save_clicked(), 每次更新 reflection 和 timeusage 的值, 并写入 txt 文件。
5. 通过已读档的两个任务指针数组显示计划、记录总览。

2.7 各部分界面模块

在 Forms 文件夹中的 ui 文件里均进行了细致的调整, 部件样式的设置, 图标图片等的插入。

三、 小组成员分工情况

1. 张子诺

主界面、计时器界面、计划与记录界面（包括提醒、保存、删除）功能, 以及这三个界面的按钮和 label 的 UI, 动图的添加, 托盘, 时间实时显示, 报告撰写。

2. 倪嘉怡

主界面、计时器界面、计划与记录界面整体 UI（背景图片、组件位置关系、字体、大小等细节), 更换背景图片、更换背景颜色、更换全局字体功能, BGM 选取和循环播放。

3. 杨斯淇

时间计划与记录完成情况对照界面的功能及 ui（包括反思、利用率填写及存档), 计划、记录总览两个页面的功能及 ui。报告排版。

四、 项目总结与反思

1. 首先, 在进行项目制作的过程中, 我们学会更多的从用户体验、方便使用的角度来添加功能, 对于时间管理软件, 用户更加注重的是学习效率、时间利用率的提高, 因此我们从各个角度为用户营造良好的环境, 选择最适合的功能。

(比如为计时过程加入 BGM 这一功能, 以 qt 自带的 piano.mp3 为例, 时长为 1m20s, 重复类似乐句片段, 且在播放完之后自动循环播放, 让大脑在潜意识层面容易预测音乐的走向, 让听者可以将注意力从音乐转移到其他事而不会感到惊讶或困惑, 更容易专注。此外, 我们还在钢琴音之外加上持续的绵密的背景白噪音, 这样纹理一致的声音覆盖物让听者拥有舒适感且更容易激发专注。因此好的 BGM 对于计时与专注是有效果的。

比如界面的初始化背景是我们精挑细选的一张淡蓝色背景, 没有任何特殊线条, 在任何界面和组件上都比较美观柔和, 希望带给用户美好的体验。)

2. 其次, 我们对项目的细节方面更加注重。比如对于选项的设计（如“选择背景图片”选项等), 我们加入了一些小图标或背景图片, 使得这些 QPushButton、QLabel 更加生动协调; 再比如在主界面背景图片的基础上, 我们又加入了一些可爱的动图, 使得整个界面更加美观; 再比如花费很长时间对各个界面 UI 的调整, 我们力求界面布局、各个组件的位置关系、字体大小等细节尽善尽美。这些细节方面的完善都是

在不断地修改代码的过程中尝试加入的，这对于我们培养更加严谨的思维有很好的效果。

3. 最后，由于是小组合作，我们也通过这个项目培养了合作协调能力和沟通能力。组长对统筹协调小组各成员工作、协助各位成员完成代码、合理安排代码进度有了更加明确的认识。小组成员则在代码合作、及时的交流沟通、代码整合与兼容性上有了更多的经验。