

Timer Lite 介绍

TimerLite 是一款轻量级的计时软件，满足记录学习、工作时长，并以统计数据的形式展现的需求。TimerLite 在设计时主要考虑了用户界面和交互的设计，尝试避免“技术含量高的程序颜值不高、用户操作不明”的现象。

App 操作说明

根据作者的设计理念，她希望用户[在阅读前先自行尝试 App 的使用](#)。软件的功能被设计得尽量简单，且操作符合直觉，理论上来说不需要太多文字辅助。如果对任何操作和功能有疑惑，App 的操作说明如下。

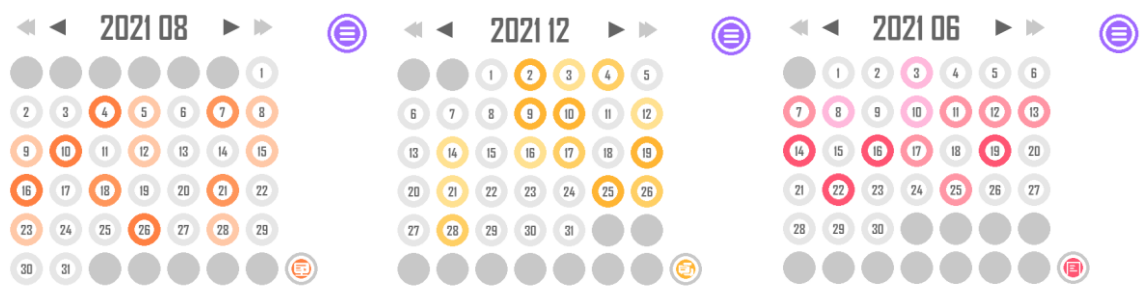
· 主界面



· 日历展示界面



日历界面其他类别效果展示如下。



App 设计理念

目前，轻量级的计时器软件逐渐普及，比如 Forrest、番茄钟、Zeitgeist 等，通过为工作计时达到集中精神、提高工作效率、规范作息时间的目的。然而，作者在使用这些软件时发现，它们有的拥有太多附加功能，反而会打扰到原本的工作；有的功能过于简洁，很难体现计时后的成就感；有的虽然功能齐全，却使用复杂，界面缺乏美感。因此，作者萌生了自己完成一款轻量级计时软件的想法。

作者希望，软件的操作符合用户的直觉，可以通过足够的视觉引导，让用户完成操作，无需过高的学习成本。同时，界面简洁美观，让用户可以欣赏自己计时的成果，也不会沉迷于冗余的功能。

作者在此也希望呼吁各位程序员/未来的程序员，在写代码的同时学一些设计，拥有基础的审美，在完成程序的同时也考虑一下用户的接受。设计的概念同时也可以包含在文档中。

App 设计亮点

1. 软件内，除数字以外，不使用任何文字。作者认为，一个优秀的“小程序”应该做到通过程序内案件图标的指引完成教学，无需太多学习成本。图标和数字作为世界通用的交流媒介，在操作不复杂的情况下，完全可以代替语言，降低交流成本。
2. 软件采用简约的几何风格，搭配明快的颜色，使程序美观度大大增加。

未来开发方向

1. 加入自定义 ColorTheme 功能，让用户为界面选择自己喜欢的颜色，同时作者提供 Night Mode 和多种配色方案选择。目前，右侧侧边栏和动画略显冗余，因为里面只有一个切换界面按钮，未来将在侧边栏增加颜色选择按钮。
2. 提供更多样的统计数据的图表类型，例如一日的饼图、一周的柱状图、一年的点阵图等等。
3. 使用数据库替换文本文档来记录计时数据。数据库能够增加查找信息的速度，并且不容易被用户无意中篡改。

以下将从代码角度介绍 Timer Lite。

资源文件说明

需要的库：pygame，其他没有额外需要 install 的

字体文件：AGENCYB.TTF

信息存储文件：record.txt

图标文件：png 格式图片若干

核心算法/数据结构说明

Timer Lite 在算法与数据结构上的优势主要包括以下几点。

1. 创建的类对象层次清晰

CLS_button, CLS_text_box – 按钮、输入框等用户交互组件

CLS_timer, CLS_calander – 时钟和日历组件，程序核心

CLS_record – 管理记录下来的时间，可以查询、保存、删改

CLS_window – 类似于 Framework 框架，管理所有组件，处理 keydown、mousedown 等事件，管理程序状态（界面切换）

2. 处理按钮时间时使用回调函数

为了提高交互组件的代码利用率，并且不让按钮组件在类内部管理其他组件的状态，避免对象“越级管理”，作者使用了回调函数，在主程序内定义点击后执行的任务，按钮类只负责执行该函数。

```
btn_play.callback_click = callback_btn_play
btn_label.callback_click = callback_btn_label
btn_switch.callback_click = callback_btn_switch
btn_tabbar.callback_click = callback_btn_tabbar
btn_timedsp.callback_click = callback_btn_timedsp
btn_calendar_label.callback_click = callback_btn_calendar_label
btn_calendar_rmonth.callback_click = callback_btn_calendar_rmonth
btn_calendar_lmonth.callback_click = callback_btn_calendar_lmonth
btn_calendar_ryear.callback_click = callback_btn_calendar_ryear
btn_calendar_lyear.callback_click = callback_btn_calendar_lyear
btn_textbox_lb.callback_click = callback_btn_textbox_lb
btn_textbox_ls.callback_click = callback_btn_textbox_ls
btn_textbox_rb.callback_click = callback_btn_textbox_rb
btn_textbox_rs.callback_click = callback_btn_textbox_rs
```

图：为按钮的回调函数赋值的代码

3. 将颜色参数定义为全局变量

全局变量的使用可以增加代码可读性，在代码中尽量减少出现不知所云的数字，而是使用颜色的名称代替，也减少了修改颜色的成本。

```

WHITE, BLACK = (255, 255, 255), (0, 0, 0)
GR1 = (240, 240, 240) # FOR THE BACKGROUND OF PRESSED/MOUSEON BUTTON
GR2 = (230, 230, 230) # FOR CALENDAR - DAYS THAT ARE NOT TIMED
GR3 = (200, 200, 200) # FOR CALENDAR - DAYS THAT DOESN'T BELONG TO THIS MONTH
GR4 = (100, 100, 100) # FONT COLOR

RED = (255, 85, 115)
ORANGE = (255, 128, 66)
YELLOW = (255, 181, 51)
GREEN = (146, 220, 55)
CYAN = (24, 218, 189)
BLUE = (54, 143, 231)
PURPLE = (160, 105, 255) # THE ORIGINAL COLOR STYLE

# from brightest to darkest
GREY_LIST = (WHITE, GR2, GR3, GR4, (0, 0, 0))
#           white      gr1      gr2      gr3      black
PURPLE_LIST = ((200, 170, 255), (180, 135, 255), PURPLE)
RED_LIST = ((255, 185, 220), (255, 150, 165), RED)
ORANGE_LIST = ((255, 200, 167), (255, 150, 91), ORANGE)
YELLOW_LIST = ((255, 225, 145), (255, 207, 100), YELLOW)
GREEN_LIST = ((185, 255, 118), GREEN, (90, 160, 45))
CYAN_LIST = ((90, 220, 200), CYAN, (8, 150, 140))
BLUE_LIST = ((140, 185, 235), (70, 145, 225), BLUE)

COLOR_LIST = (PURPLE, RED, ORANGE, YELLOW, GREEN, CYAN, BLUE)
CLD_COLOR_LIST = (PURPLE_LIST, RED_LIST, ORANGE_LIST, YELLOW_LIST, GREEN_LIST, CYAN_LIST, BLUE_LIST)

```

图：将颜色参数定义为全局变量的代码

4. 画图时通过计算的方式得出坐标

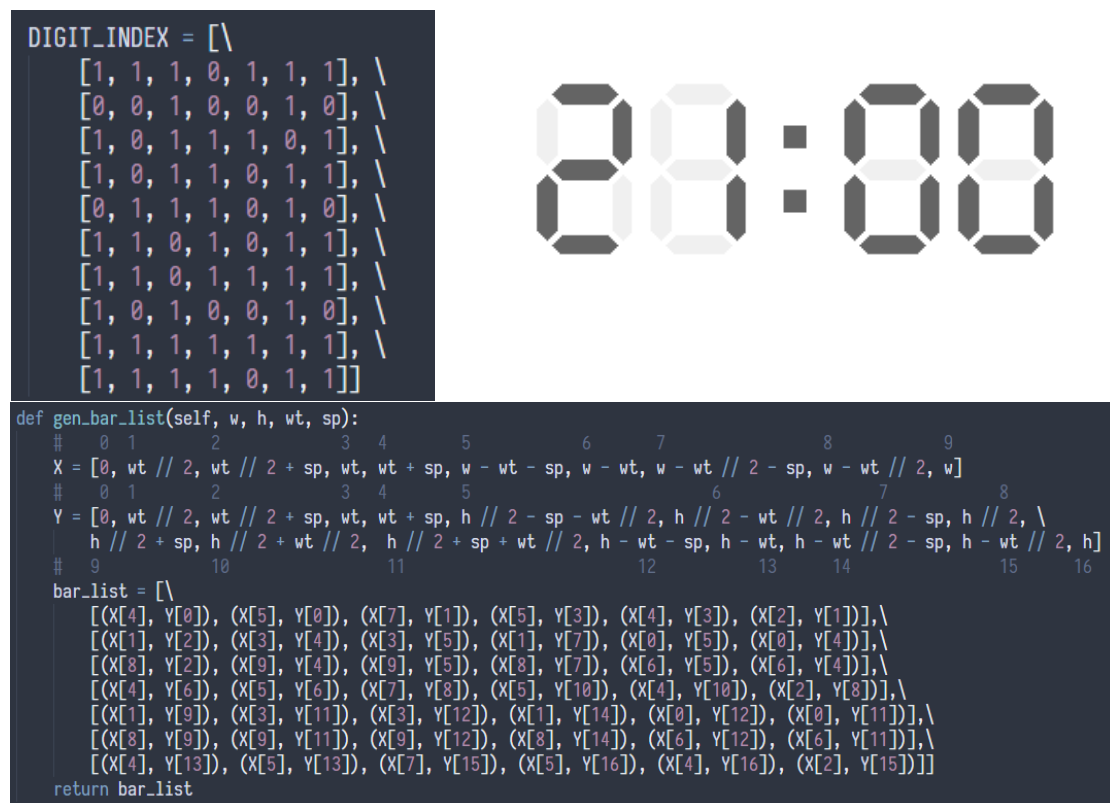
在程序中，需要进行大量几何形状的绘图，将所有坐标点都用变量 x 倍数的方式表示，能够增加可读性和可理解性。

```

if flag == 1: # play
    pygame.draw.circle(surf, clr, (r, r), r, w)
    pygame.draw.circle(surf, bgclr, (r, r), r - w)
    pygame.gfxdraw.aapolygon(surf, [(r * 0.7, r * 0.6), (r * 0.7, r * 1.4), (r * 1.5, r)], clr)
    pygame.draw.polygon(surf, clr, [(r * 0.7, r * 0.6), (r * 0.7, r * 1.4), (r * 1.5, r)])
elif flag == 2: # pause
    pygame.draw.circle(surf, clr, (r, r), r, w)
    pygame.draw.circle(surf, bgclr, (r, r), r - w)
    pygame.draw.rect(surf, clr, (r * 0.7, r * 0.5, r * 0.2, r))
    pygame.draw.rect(surf, clr, (r * 1.1, r * 0.5, r * 0.2, r))
elif flag == 3: # stop...
elif flag == 4: # color choice
    pygame.draw.circle(surf, clr, (r, r), r, w)
    pygame.draw.circle(surf, bgclr, (r, r), r - w)
    if fclr == None:
        fclr = clr
    pygame.draw.circle(surf, fclr, (r, r), r - w * 2)
elif flag == 5: # leftplay...
elif flag == 6:...
elif flag == 7:...
elif flag == 8:...
elif flag == 9:...
elif flag == 10:...
elif flag == 11: #intentionally lefted blank for words...

```

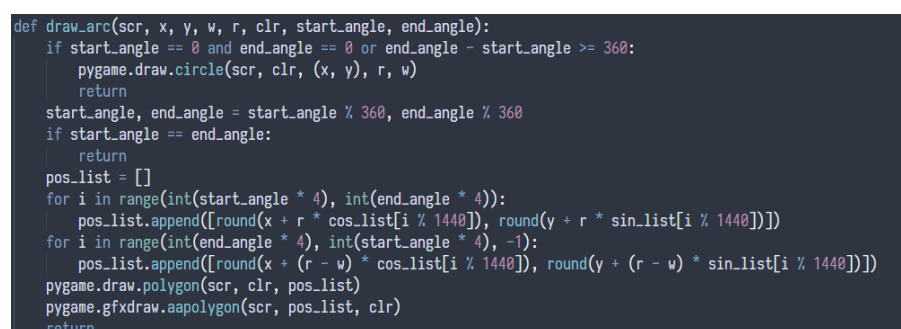
图：根据按钮半径 r 绘制按钮图案的代码



图： 左上 - 使用二值图的方式完成数字信息储存的代码
 下 - 代码中定义一遍所有需要用到的坐标点，再用列表的 index 调用，缩短代码
 右上 - 最终绘制效果

5. 自定义 draw_arc 函数绘制带有宽度的、抗锯齿的弧线

在 pygame 中，没有绘制有宽度的弧线的函数。同时，抗锯齿的图形无法填充内部。作者利用三角函数计算出一个近似于圆的多边形的坐标，再绘制一个抗锯齿的边缘和一个不抗锯齿的内部填充图形，完成弧线绘制。



图： 左 - draw_arc 函数的代码
 右 - 弧线效果（紫色部分）