

壹、前言

為了解決讀書時無法持之以恆、不專心的壞習慣，我們找到了近期許多人推薦以「番茄鐘工作法」為基礎的應用程式，但是在使用過後總覺得功能不足。例如有的番茄鐘 App 有計畫表的功能，卻因沒有日曆而造成排定行程的困擾；有的則是功能應有盡有卻都需要付費才能使用。每一個類似的應用程式都沒有同時符合我們的需求，因此我們想嘗試製作一個改良版的時間規劃管理應用程式來解決問題。

一、研究動機

每當坐在書桌前寫作業讀書時，總是寫了一兩道題目卻又拿起手機來玩，讀不到半小時又想站起來動一動。這種情況也是許多人在嘗試集中精神處理眼前的工作時常面臨的問題，課業壓力繁重的我們更為此懊惱不已。為了解決這樣不專心無法持之以恆的壞習慣，發現了一種 25 分鐘專心、5 分鐘休息的「番茄鐘工作法」，除了用在工作，也可以在讀書時使用。

市面上有許多這樣的應用程式是以番茄鐘工作法為基礎的，但是我們對功能的要求越來越多，因此想要自己嘗試製作出一款以番茄鐘為基礎的時間規劃管理應用程式，來解決時間管理和功能不足的問題。

目前有許多 App 開發軟體，其中 App Inventor 為最普遍且容易上手的軟體，但僅適用於 Android 系統，因此，本研究使用了可跨平台的 Unity 來進行開發。

二、研究目的

- (一) 學習 Unity 開發應用程式的技能
- (二) 了解番茄鐘時間法在學習上的效率
- (三) 設計一款具有多元功能，例如白噪音，適合各領域年齡層的時間管理應用程式
- (四) 加上計分功能，與朋友比賽專注時長並賺取積分達到良性競爭，提高學習效率
- (五) 藉由本研究設計的應用程式提升在學習狀態的專心度，養成好習慣並提高效率
- (六) 本研究製作的 APP 與市面上 APP 的功能分析與比較

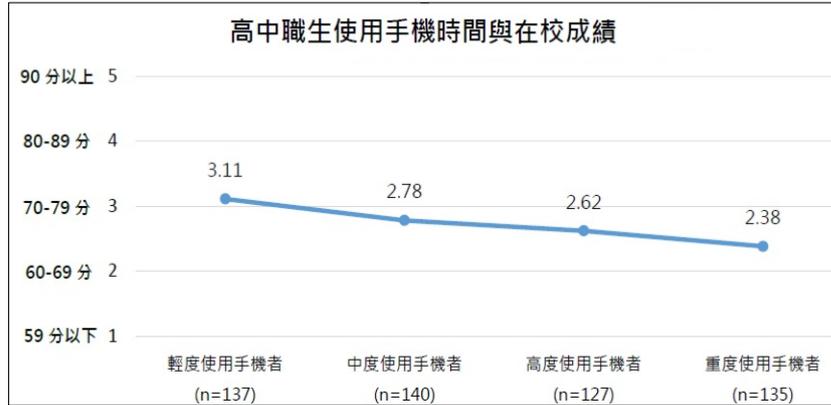
貳、文獻探討

一、青少年的時間管理

根據科技部傳播調查資料庫第二期第四次（2020）青少年調查，內容指出在接受調查的台灣高中職生中，結果顯示高中職生一周有六天皆會使用手機上網，而他們使用手機的日子在學習、工作的時間約 1.5 小時，在娛樂休閒時間則將近 4 小時，可見得，高中職的學生們用手機花了不少時間在休閒娛樂上。

由下圖，我們可以看出手機使用的情況會反映在成績上，平均上愈是重度的使用者，成績就愈差。可見得，若不能好好管理使用手機的時間，將會對自己的課業、學習產生不良影響。

圖一：高中職生使用手機時間與在校成績統計圖



圖一來源：臺灣傳播調查資料庫

二、番茄鐘工作法對學習效率的影響

蕃茄鐘工作法是一種時間管理方法，執行它有五個步驟

- (一) 決定目標任務
- (二) 設定計時器到 x 分鐘(通常是 25 分鐘)
- (三) 持續執行該任務直到時間到
- (四) 短暫休息數分鐘
- (五) 重複 n 次(通常是 4 次)步驟 2~4 後，休息 15~30 分鐘

這樣的時間管理方法優點在於能持續地維持人的專注力，休息時間能夠讓頭腦消化、吸收剛才 25 分鐘的工作內容，提升工作/學習的效率。缺點則是在 25 分鐘的工作時間內，是不能被打斷的，這會影響專注力，所以要挑在不會被打擾的時間進行。因為這項方法能維持人的專注力，所以能夠為學習帶來很好的幫助。

三、研究證實番茄鐘工作法有效提升效率

《Educational and Developmental Psychologist》刊載研究證實：澳洲雪梨大學召集 72 名大學生受試者，要求他們在 20 分鐘內參加一堂預計耗盡受試者的注意力水平的數學課程，並將其分為 3 組人馬：一組在課程期間進行了 5 分鐘的隨機休息時間，第二組則是休息 5 分鐘，但被要求觀看一段與雨林生態有關的影片，而第三組完全沒有休息，直接進入下一堂數學課。

課程結束後，所有學生都完成了一份關於定向注意力水平的自我報告問卷，以及課後的隨堂測驗。分析結果表明：第一組、第二組受試者相較第三組，展現出的自我報告專注度更高；前兩組的隨堂測驗分數，明顯也比第三組來得出色。

另外也發現，因挑戰性任務而耗盡的注意力，可以通過非結構化休息時間來恢復，且休息時接觸自然環境，無論是實際接觸或透過觀看虛擬影片，都對恢復注意力、提高答題正確率有所助益。

圖二：澳洲雪梨大學研究結果呈現



圖二來源：研究者自行繪製（參考自潮健康：「番茄鐘工作法」真能提升工作效率？）

四、時間管理應用程式

基本的番茄鐘功能有計時器，白噪音等功能；其餘像是任務清單和加入小組功能視應用程式不同而提供。

(一)計時器功能：以預設的番茄鐘工作法：工作 25 分鐘、休息 5 分鐘最為基礎，而有些應用程式則會提供正、倒計時供使用者選擇。

(二)白噪音：在一般人可輕易接受的範圍內，一段頻率不同但強度保持一致的聲波，例如雨聲、浪濤聲等等。白噪音有屏蔽或弱化其他聲音的效用，並因此能安撫神經，讓人鎮定下來，使使用者能夠更專注在工作狀態；

(三)小組功能：讓使用者之間能夠互相競爭，利用讀書時長累積積分並排名，使用者便能從中獲得成就感，進而提升工作效率。

在市面上多款的應用程式中，我們選擇了「Forest」和「專注清單」做比較，如表一所示。除了參考網路上的使用排名及身邊好友推薦，也基於他們以不同面向及方式讓使用者能夠體驗不同專注方式讓我們去比較差異及成效，並從中參考研發出給更多年齡層及各領域的人使用。

表一：「Forest」和「專注清單」App 比較

項目	圖示	適用平台	計時器	白噪音	任務清單	計分排名	加入小組	正/倒數計時
Forest		Android IOS	✓	更多鈴聲須付費	付費解鎖			✓
專注清單		Android IOS	自訂義須付費	更多鈴聲須付費	付費解鎖限制	✓	✓	

表一來源：本研究自行整理

功能上而言，雖然「專注清單」已經相當齊全，但在任務清單的操作方面有些複雜之處，而且許多功能也是要另外購買才能使用，對於一般人使用者而言不是那麼的友善；「Forest」雖然很適合給容易分心的人們使用，但在排定日程方面需要額外的付費解鎖，這可能會導致使用者無法方便地記錄預定的待辦事項。

除了市面上的應用程式，吳哲亨(2014)在蕃茄鐘時間管理法之應用及其於 Android 系統開發平台之實現的研究中，作者基於一篇天下雜誌文章，名為“番茄鐘工作法！改變習慣作好「時間 整理」”，進而構想出利用了 App Inventor 開發一款番茄鐘時間管理程式，作者的研究重點除了在製作時間管理應用程式，還有使用“ActivityPackage”元件呼叫其他應用程式，讓撰寫者結合其他應用程式，增加使用上的便利。

綜合上述，我們了解到番茄鐘所帶來的效益，以及多數 APP 有付費功能的限制，因此，本研究希望設計出一款能夠對於一般學生更優質、更便利的時間管理應用程式。

參、研究方法

一、研究流程

本研究為了更有效率完成待辦事項，著手研究開發符合自己需求時間管理應用程式。尋找了相關的文獻後，試著以 Unity 設計一個以蕃茄鐘為基礎的 APP。

圖三：研究流程圖

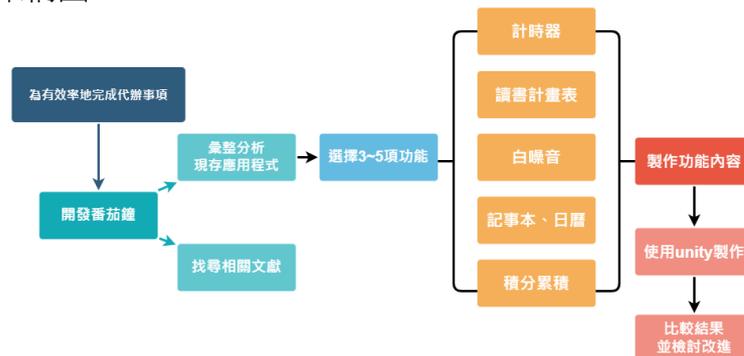


圖三來源：本研究自行繪製

二、研究架構

目前雖有許多與蕃茄鐘相關的時間管理應用程式，但部份功能仍需付費，因此本研究自行設計了一款涵蓋了讀書計畫表計時器、白噪音、日曆、積分等功能的 APP

圖四：研究架構圖



圖四來源：本研究自行繪製

三、研究工具

在文獻的查找中看到賴識曉(2023)的研究中使用了 Unity 來開發應用程式，加上我們之前有使用過 Unity 開發過小遊戲，瞭解到 Unity 能夠打造出更順暢、易用的介面，儘管 App Inventor2 的使用難度較 Unity 低，我們決定試著使用 Unity 來做開發我們所需的番茄鐘 App，期望能夠實現比 App Inventor2 更精緻的功能和效果。

肆、研究分析與結果

一、Unity 開發時間管理程式

我們的 App 程式主要五項功能，分別說明介紹如下：

(一) 系統功能

1.計時器

利用番茄鐘工作法的專心 25 分鐘、休息 5 分鐘為基礎的計時器，讓使用者記錄自己專心的時間，也不忘停下來休息喘口氣。若是不習慣這樣的模式，還可以自訂專心和休息的時間。

2.讀書計畫表

使用者可以排定每天的日程表，像做任務一樣一件一件完成，方便確認完成即未完成的任務，以幫助使用者更好地管理工作和學習。

3.記事本、日曆

結合了記事本和日曆功能，使用者可以紀錄重要日期和事件，例如考試日期、作業截止日期等。也可以利用記事本功能寫日記、記錄重要訊息。

4.白噪音

使用此應用程式計時時，可以用螢幕顯示的倒數計時，還能選擇播放適合自己的白噪音，營造更有氛圍的工作環境。

5.積分累積

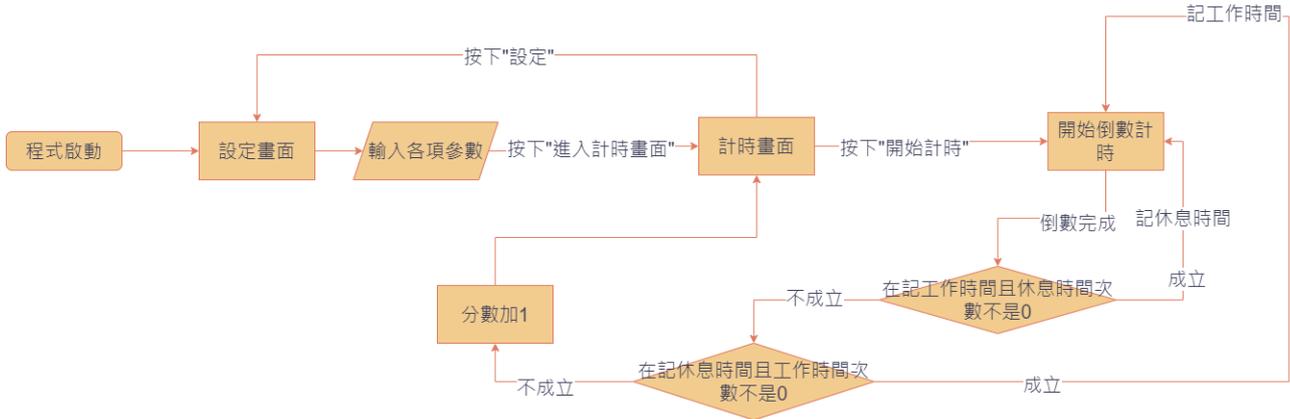
使用計時器完成任務後，使用者可以獲得專心積分，這可以提供一種獎勵機制，鼓勵使用者保持專心；還可以與夥伴一起使用這個應用程式專心讀書或工作，並通過積分互相競爭和鼓勵，提高效率。

(二) 程式碼解說

1.計時器

當按下啟動按鈕，可以進行計時器及音樂播放功能，計時器暫停，音樂播放也同時暫停。由於計時器是基於番茄鐘設計，會先計時一段工作的時間，接著計算休息時間、重複數次，計時器的運作架構如圖五所示。若使用者想定的時間可能不一樣，也可以自訂，例如預設為工作 25 分鐘、休息 5 分鐘、重複 3 次。如果讓計時器跑完，就可以獲得一分，且分數不會因為重啟程式就被重置。下圖六、圖七為計時的主要函數。

圖五：計時功能架構規劃圖



圖五來源：本研究自行繪製

圖六：計時函式 1

```
IEnumerator StartCountdownCoroutine(int Total,int BeginTime)
{
    Debug.Log("StartCount");
    int Left;
    while (Total - Past > 0)
    {
        NowTime_f = Time.time;
        NowTime_i = Mathf.FloorToInt(NowTime_f);
        Past = NowTime_i - BeginTime;
        Left = Total - Past;
        Minute=Left/60;
        Second=Left%60;
        Timer.text = string.Format("{0}:{1}", Minute.ToString("00"), Second.ToString("00"));
        yield return null;
    }
    IsCount = false;
    Past=0;
}
```

圖六來源：本研究 Unity 內截圖

圖七：計時函式 2

```
if(Work && RestTime>0){
    Total_Time=Rest_m*60+Rest_s;
    RestTime--;
    StartTime_f = Time.time;
    StartTime_i = Mathf.FloorToInt(StartTime_f);
    IsCount=true;
    Work=false;
    yield return StartCoroutine(StartCountdownCoroutine(Total_Time, StartTime_i));
}
else if(Work && WorkTime>0){
    Total_Time=Work_m*60+Work_s;
    WorkTime--;
    StartTime_f = Time.time;
    StartTime_i = Mathf.FloorToInt(StartTime_f);
    IsCount=true;
    Work=true;
    yield return StartCoroutine(StartCountdownCoroutine(Total_Time, StartTime_i));
}
else{
    WorkTime=Total_WorkTime;
    RestTime=Total_RestTime;
    Data.Score++;
    PlayerPrefs.SetInt("Score",Data.Score);
    PlayerScore.text=Data.Score.ToString()+"分";
    Control_P_C(false);
    ADS.Stop();
}
```

圖七來源：本研究 Unity 內截圖

由圖六顯示，IEnumerator 函式進行呼叫需使用 StartCoroutine 程式，並且要引入 Past、Time.time 兩個數值。變數 Past 儲存從開始計算時間後，至目前經過了多久；Time.time 可以取得當前的時間，用 Mathf.FloorToInt() 可以將括號中 float 形式變數轉為 int，將要計算的時間扣除經過的時間即為剩下的時間，因為在這使用的時間單位都是秒，所以將他除以 60 就可以轉為分鐘，除 60 取餘數就是秒數。

圖七是在完成一次計時會被執行到的內容，布林值 Work 代表現在是否在計算工作的時間，WorkTime 及 RestTime 分別代表剩下幾次工作的次數和休息的次數要算，如果現在是在計算工作的時間，且要計算休息的次數還大於 0，代表接下來要記休息的時間，於是會設定好所需的參數，並再次呼叫這個函數來繼續計時，反之亦然。當休息和工作的時間都計算完後，就會重置資料，並把分數加上 1，這裡的 Control_P_C() 是控制暫停、撥放鍵的顯示，當傳 false 時會將其隱藏，最後把音樂停止，就完成一段計時了。

yield return null; 是用來避免程式阻塞，導致系統崩潰的程式碼，可以使這個程式停頓一下下，避免迴圈無限循環消耗過多資源。

2.月曆程式碼

創建一個儲存日期的 class，裡面有日期、顏色、對應物件的參數，程式碼如下圖八、圖九所示。創建一個 Day 型態的 list，第一次載入時先一個一個按照日期給定日期、顏色、對應物件等參數後，加入 list 中，下圖程式會先被給定年和月份，startDay 是該月第一天的星期代號，從禮拜日到禮拜天分別是 0~6，endDay 是該月有幾天的變數。利用這些進行運算即可給定每個所需以的參數。

圖八：月曆程式碼 1

```
public class Day
{
    3 references
    public int dayNum;
    1 reference
    public Color dayColor;
    3 references
    public GameObject obj;
    3 references
    public Day(int dayNum, Color dayColor, GameObject obj)
    {
        this.dayNum=dayNum;
        this.obj=obj;
        UpdateColor(dayColor);
        UpdateDay(dayNum);
    }
}
```

圖八來源：本研究 Unity 內截圖

圖九：月曆程式碼 2

```
if (days.Count==0)
{
    for (int w=0;w<6;w++)
    {
        for (int i=0;i<7;i++)
        {
            Day newDay;
            int currDay=(w*7)+i;
            if (currDay<startDay || currDay-startDay>=endDay)
            {
                if(currDay<startDay)
                    newDay=new Day(lastMonth+(currDay-startDay),Color.grey,weeks[w].GetChild(i).gameObject);
                else
                    newDay=new Day(currDay-startDay-endDay, Color.grey,weeks[w].GetChild(i).gameObject);
            }
            else
                newDay=new Day(currDay-startDay,Color.white,weeks[w].GetChild(i).gameObject);
            days.Add(newDay);
        }
    }
}
```

圖九來源：本研究 Unity 內截圖

(三) 成果展示

1. 計時器

圖十是 App 內設定的畫面，使用者可以在輸入框中填入預想的時間倒數。使用者可於輸入框中輸入自訂的時間，如圖十一所示，同時在左上角也會出現目前的積分累計。當在設定畫面點擊"進入計時畫面"按鍵，就會進到圖十二的計時畫面，一開始畫面上分別有兩個按鈕，設定是回到設定畫面，開始計時會依照設定時填入的參數開始計時。開始計時後，會出現兩個按鈕，分別是繼續與暫停鍵，位於「開始計時」按鈕下方兩個按鈕(它們是藉由將 App 運行速度設為 1 和 0 來實現)。

2. 白噪音

圖十三為白噪音功能，使用者可選擇自己喜歡的音樂播放。音樂來源為 YouTube 無版權音樂。

圖十



圖十一



圖十二



圖十三



圖片來源：本研究 APP 截圖

3.計分累積

上方會顯示現在的計時是第幾次的工作/休息，在計時結束時，畫面中間顯示的分數會加 1，以此作為積分累計。

4.日曆、記事本、讀書計畫表

當點擊"筆記"按鈕，會跳轉到圖十四的畫面，日曆會顯示當前月份，點擊日曆上日期可跳轉到記事本畫面，如下圖十五所呈現。記事本使用的如圖十六、十七，使用者可在輸入欄中輸入代辦事項或是讀書計畫，可以提醒使用者那天需要做的事。

圖十四：自製蕃茄鐘日曆畫面



圖十四來源：App 內截圖

圖十五：自製蕃茄鐘筆記畫面



圖十五來源：App 內截圖

圖十六：自製蕃茄鐘記事本使用畫面



圖十六來源：App 內截圖

圖十七：自製蕃茄鐘讀書計畫提醒畫面



圖十七來源：App 內截圖

伍、結論與建議

一、結論

本研究成功的使用 Unity 開發了符合我們需求的 APP，但在實作過程中也並非一路順遂。像是最開始計時的部分沒有利用排程，直接使用了 void 函式，導致程式碼直接卡死。而最後也留下了一些問題：在場景切換時，會需要一點時間讀取，但畫面會停在原本的畫面，導致使用者可能以為沒按到按鈕。另一個問題是這樣的 App，在完成後，容量達 100 多 Mb，遠比其他蕃茄鐘 App 的空間用量還大，這或許是使用 Unity 開發所產生的問題。

我們也與其他兩款 APP 進行了比較，如表二所示。我們的 APP 功能多於他人論文

研究，坊間著名的 APP 功能看似完整，但某些功能需付費才能使用。我們也對於自己的設計成果做了 SWOT 分析，如圖十八所示。本研究開發的 APP 對於一般學生使用上較為友善，沒有功能付費解鎖的問題，也能讓使用者自行定義專注與休息時間，較為彈性。然而應用程式檔案大、尚未有優美的 UI 設計，要與坊間的時間管理 APP 做特色比較，尚有很多進步空間。

表二：本研究與其他時間管理應用程式比較表

	自製番茄鐘	Forest	吳哲亨(2014)番茄鐘
主畫面	設定時間及音樂設置畫面	計時畫面	代辦事項
專注時長	能夠自訂義專注及休息的時間，或使用預設基礎番茄鐘	能夠自訂義專注及休息的時間，或使用預設基礎番茄鐘，並備有正、倒計時	基礎番茄鐘預設時間
代辦事項	以文字為主	須付費，以文字為主	可使用照片、相機功能命名代辦事項
計時按鈕	工作時間結束後直接接休息時間	僅計時專注時間，不會計時休息時間	分開按鈕執行
獎懲功能	完成一次番茄鐘累積一點積分；並無懲罰	完成一次番茄鐘依照時長賺取金幣；若跳離應用程式，樹木將枯萎	無
社交功能	無	需付費，可跟其他使用者排名增加競爭力	無
互動	僅能累積積分，無特殊互動	邊專注邊種樹，使用者能獲得成就感	無
其他功能	無	會有鼓勵話語以及阻擋外來程式提醒提示	無

表二來源：研究者自行繪製

圖十八：本研究 App 之 SWOT 分析



圖十八來源：研究者自行繪製

二、建議

本研究的 APP 雖然在功能方面已具完整性，集結了我們想要的功能，有效的滿足使用者的需求，但仍有許多不足的地方可再進行研究改進。以下除了分享我們在製作過程中遇到的困難與解決方式，也建議了幾項可讓目前 APP 再行優化的方向。

(一) 開發應用程式部份

- 1、Unity 一般在執行時只會有一個執行序，要注意和時間相關的迴圈，需要使用排程，才不會讓程式卡死。在程式的大小寫上需要注意，大小寫弄錯就會沒辦法執行，所以最好在編輯器上裝上 Unity 的插件，能減少語法的錯誤。
- 2、IEnumerator 型態的函式在呼叫時必須使用 StartCoroutine (欲呼叫函式) 才能夠順利呼叫。
- 3、將目前這套用 Unity 開發的功能，使用 APP Inventor 設計，釐清 APP 容量過大的問題是否與開發程式的選擇有關。

(二) APP 設計部份

- 1、在 APP 讀取時間上的問題，未來可以新增進度條來改善使用品質
- 2、目前僅有功能性，在外觀上尚未有設計感。未來可再著墨研究於 UI 設計使用滿意度調查
- 3、設定多個主題性並設定相關的互動性活動，可讓使用者依照喜好做不同的 APP 畫面情境做選擇，藉由互動功能增加趣味性及使用率。

陸、參考文獻

吳哲亨 (2014)。蕃茄鐘時間管理法之應用及其於 Android 系統開發平台之實現。私立光華大學：學術論文。

賴識曉 (2023)。自動化檢測整合-以 Unity 開發 APP 為例。私立逢甲大學：學術論文。

林雨蓓 (2023 年 3 月 13 日)。白噪音助睡眠？有何效果、副作用？白噪音 APP 推薦。
<https://www.commonhealth.com.tw/article/87833>

陳思為。(2023 年 2 月 1 日)。十大時間管理 APP 推薦排行榜。
<https://tw.my-best.com/115697>

Dean (2024 年 1 月 22 日)。番茄鐘工作法是什麼？5 款番茄鐘 APP 推薦[部落格文章]、
<https://deanlife.blog/pomodoro-technique/>

傳播調查資料庫 (2020)。新傳播科技與生活延伸。
https://srda.sinica.edu.tw/datasearch_detail.php?id=3311#abs

潮健康 (2023 年 7 月 14 日)。想讓工作、學習更有成效？正確使用「番茄鐘工作法」、做對七件事有效紓壓。
<https://reurl.cc/j3Y7dn>