

Assignment 5

This assignment is based on handouts written by Jerry Liao, Shang-Chun Tai
Image credit: devclass.com



這份作業將使用由史丹佛老師 Sam Redmond 所撰寫的 campy 包裹，來讓你學習 Python 繪圖、動畫製作及遊戲製作的基礎觀念！其中，滑鼠偵測（MouseEvent）的使用，同時也是下一份作業 — 打磚塊遊戲的重要基石。

透過這兩份作業的練習，你也會更熟悉電腦科學裡非常具有挑戰性的「class & object」概念：

- 每個繪圖元素都是一個「object」
- 每個可以製造繪圖元素的工廠都是一個「class」

如果過程卡關歡迎各位向助教詢問！也非常鼓勵同學們互相討論作業之概念，但請勿直接把 code 分享給同學看，這很可能會剝奪他獨立思考的機會，並讓他的程式碼與你的極度相似，使防抄襲軟體認定有抄襲嫌疑

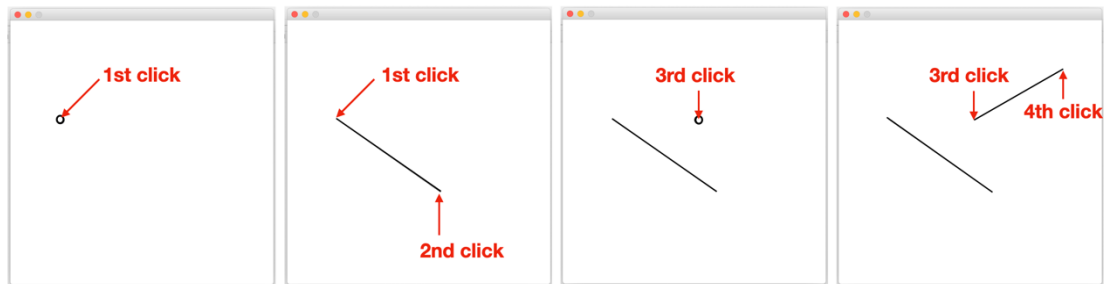
Problem 1 – draw_line.py

在上課範例中，我們成功地寫出了「小畫家」應用程式裡面的畫筆！現在讓我們來試試新功能：不斷在視窗上，畫出「兩點一線」

詳細過程如下：

奇數次點擊的位置，是這條線的起點。此時，會以滑鼠點擊的位置為圓心，畫出一個半徑為常數 SIZE 的 空心圓；偶數次點擊的位置，是這條線的終點。此時，空心圓會消失，並畫出一條直線。

請注意，整個過程的滑鼠動作只有點擊（click）視窗，沒有拖拉（drag）。程式執行畫面的範例請見下圖（紅色的箭頭與文字是我們後製上去的，真正的程式畫面只有圓形與線條）



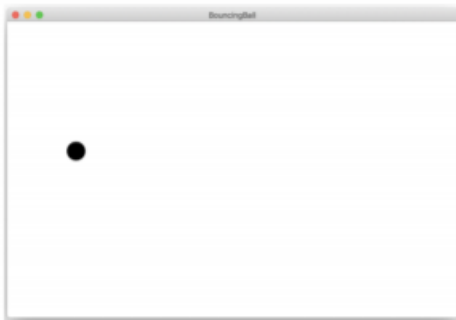
- 左一圖片是使用者第一次點擊視窗時，畫出一個空心圓
- 左二圖片是使用者第二次點擊視窗時，畫出一條直線，空心圓消失；線條起點為先前空心圓的圓心，終點為滑鼠第二次點擊時的座標
- 左三圖片是使用者第三次點擊視窗時，畫出另一個空心圓
- 左四圖片則是使用者第四次點擊視窗，畫出的新的直線。在程式終止前，只要使用者持續點擊，電腦就可以按照這個規律，持續畫出直線！



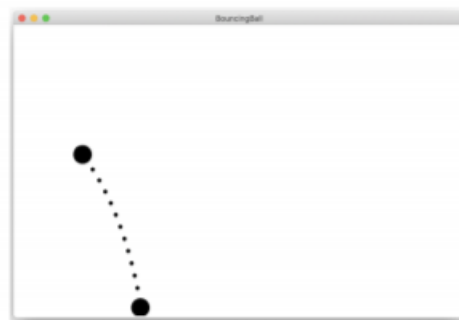
Problem 2 – bouncing_ball.py

最後一題會是下週「打磚塊」魔王作業的前哨戰！各位同學將以電腦程式，模擬一顆球的彈跳過程。圖(一)～圖(六)為球在墜落反彈時的分解示意圖。（虛線是後製上去的，以輔助題目解釋）

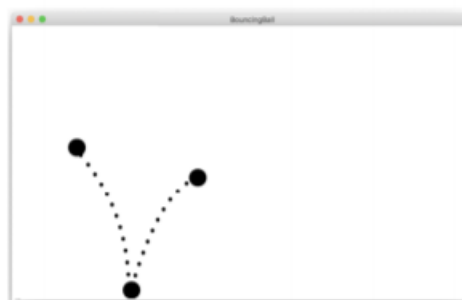
也歡迎參考影片示範：<https://youtu.be/cSSPtDkDwoM>



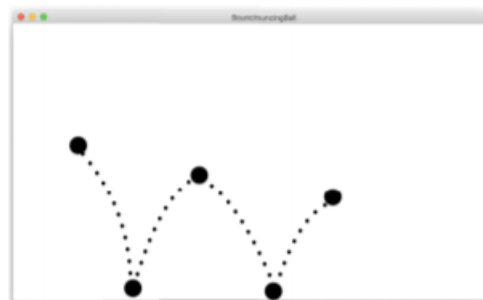
圖(一) 球的起始位置



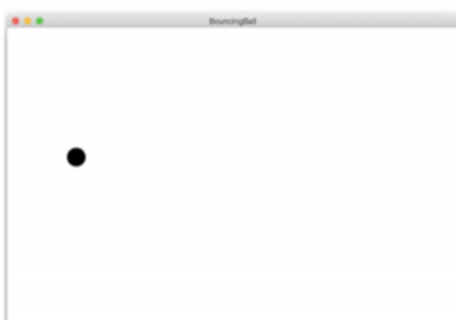
圖(二) 第一次與地板反彈之示意圖



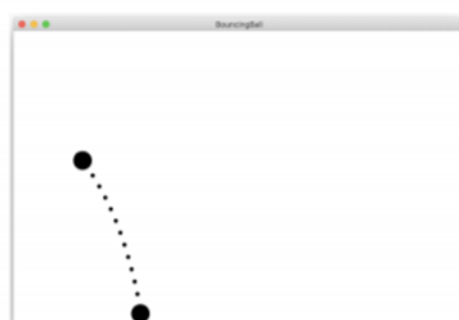
圖(三) 第一次反彈至最高點示意圖



圖(四) 第二次反彈至最高點示意圖



圖(五) 球超出視窗後回到原位示意圖



圖(六) 點按視窗，球又再次墜落

Constants 介紹（請務必使用下方的每一項）

- VX # 球的水平速度
- DELAY # 動畫停格多少毫秒（ms）
- GRAVITY # 重力加速度；每一圈 while loop 垂直速度要加上的數值
- SIZE # 球的直徑
- REDUCE # 每一次反彈時，在垂直速度所剩之比例
- X_START # 球的起始 x 座標
- Y_START # 球的起始 y 座標

程式執行後，球會停留（漂浮）在起點。待使用者點擊視窗時，才會開始往下掉。一但往下掉之後，球的水平速度保持 VX 即可。但垂直方向由於受到重力影響，往下時球速會越來越快！因此，在每一輪更新畫面時，請對垂直方向的速度加上重力加速度（GRAVITY），來模擬受球越掉越快的畫面。

接著，當球與地板碰撞時，球會往上彈，垂直方向的速度變為負值。並且，為了模擬現實中的碰撞現象，往上彈的高度會隨著碰撞的次數越多，而變得越來越低（因為有一些能量損失）。「垂直速度」會因為每次碰撞而變成「碰撞前的 90%（也就是我們定義的常數 REDUCE）」。

當整顆球離開視窗最右側時，請讓球回到原起點位置並保持不動，待使用者點擊後，球才會再度開始往下掉。

請注意：

- 球在彈跳過程中，不會受到使用者滑鼠點擊影響。換句話說，滑鼠點擊只有讓球從靜止狀態往下掉的功用。
- 在 **球超出右側視窗 3 次 後**，球將回到原起點位置，並保持不動。這時無論使用者再怎麼點擊視窗，球都不會再往下掉落。
- 不需計算球在超出右側視窗前共彈跳幾次，動畫順暢即可。

這題最困難的地方莫過於動畫要在「使用者點擊後，才能開始執行」。同學可以思考看看，什麼樣的設計可以產生類似「閘門」的功能，**讓程式停留在某種否定的狀態，並在使用者點擊（改變狀態）後，打開開關？**

在現實中，「球不會移動到地板以下的位置」；但在動畫過程中，球是有可能「位在地板之下」的！在看分解示意圖或影片示範時，球其實可能有掉到地板之下，只是因為速度太快、pixel 太小，我們肉眼看不出來！若同學遇到球卡在地板上下震動，或球掉到地面下消失的情況時，這有可能是因為球在反彈時彈不回原本的高度（因為只會彈回 90% 高度）。如果發生這種情況的話，可以想想：我們是否可以確保當球碰到地板後，垂直速度一定是向上的，讓球可以繼續往上到該次彈跳的最高點？

Problem 3 – clear_bricks.py

在上課範例中，我們試著用滑鼠點擊去模擬打地鼠的遊戲，現在我們要寫一個「接磚塊」的程式，磚塊會從天花板的隨機位置以隨機的速度落下，並且玩家要在磚塊掉到地上之前用滑鼠點擊來把他接住，如果玩家漏接了三次程式就會結束。

歡迎參考示範影片：https://youtu.be/29pU_dbBSp4

Constants 介紹（請務必使用下方的每一項）

- WINDOW_WIDTH # 視窗的寬
- WINDOW_HEIGHT # 視窗的高
- DELAY # 動畫停格多少毫秒（ms）
- SIZE # 磚塊的邊長
- MIN_Y_SPEED # y速度的最小值
- MAX_Y_SPEED # y速度的最大值

程式執行後，磚塊會從視窗的最上方開始向下落，磚塊開始落下位置的y座標為0，x座標為0到視窗寬度之間的任何整數（請注意：磚塊不能被視窗邊界切到），落下速度則會是從MIN_Y_SPEED到MAX_Y_SPEED的隨機整數。

在磚塊落下的過程中，一旦滑鼠點擊到磚塊，就會將磚塊移除，並再次隨機生成一個磚塊從視窗的最上方往下落。假如磚塊直到落地時，都沒有被玩家點擊移除，則磚塊完全離開視窗的那一刻，會再次隨機生成一個新的磚塊從視窗的最上方落下。若是磚塊沒有被點擊移除而掉落至地面3次，程式便會結束，視窗也不應再有磚塊落下。

以下幾點請同學們注意：

- 程式一開始執行，**立刻**就會有磚塊落下。
- 請避免生成被視窗邊界切掉的磚塊，每一顆磚塊在落地前都要是**完整**在視窗當中的。
- 每次**只會有一個**磚塊落下，只有在落下中的磚塊被點擊移除或磚塊落至地面後，才會再生成下一個磚塊。
- 若**磚塊落地3次**，則不會再有磚塊落下。
- 您的程式最後應該是**正常終止**的（exit code為0）

評分標準

Functionality - 程式是否有通過我們的基本要求？程式必須沒有 bug、能順利完成指定的任務、並確保程式並沒有卡在任何的無限迴圈（infinite loop）之中。

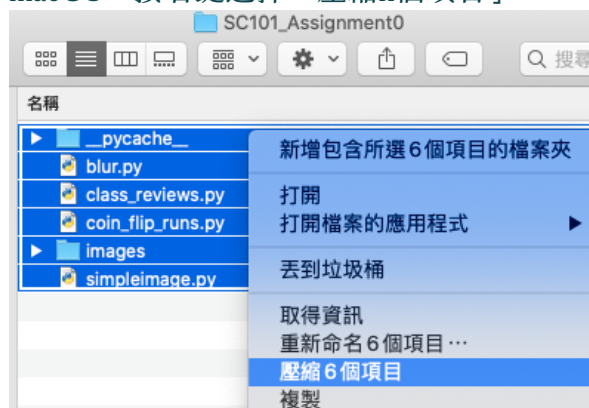
Style - 如同在課堂上所說，好的程式要有好的使用說明 (comment)，也要讓人一目瞭然，這樣全世界的人才能使用各位的 code 去建造更多更巨大更有趣的程式，因此請大家寫出精簡扼要的main()程式概要、function comments和單行註解。

作業繳交

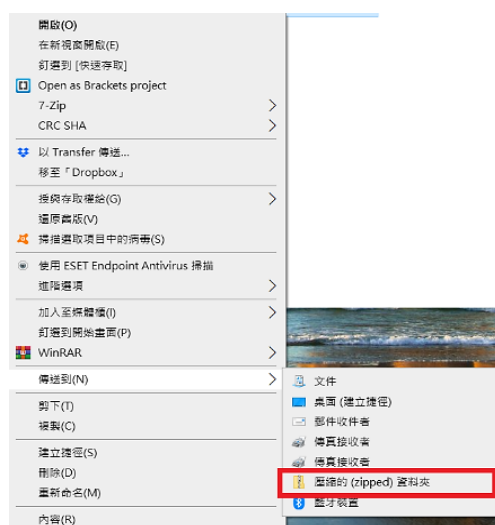
恭喜您完成 Assignment5！請同學於作業繳交期限前，依照下圖將您完成的作業的下載連結上傳至社團提供的作業繳交表單。

1. 以滑鼠「全選」作業資料夾內的所有檔案，並壓縮檔案。請見下圖說明。

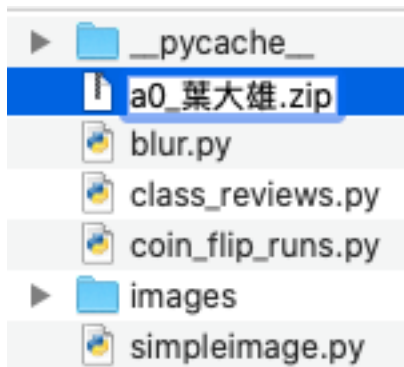
macOS：按右鍵選擇「壓縮n個項目」



Windows：按右鍵選擇「傳送到」→「壓縮的(zipped)資料夾」



2. 將壓縮檔(.zip)重新命名為「a(n)_中文姓名」。如：
assignment 0命名為a0_中文姓名;
assignment 1命名為a1_中文姓名; …



3. 將命名好的壓縮檔(.zip)上傳至new E3 作業繳交區