第二關：舟山河

**主題與遊戲介紹**

有鑒於台大旁的舟山路，每逢下雨及會淹水成為舟山河，我們決定模仿經典的躲避障礙物的遊戲，來製作一款在舟山河上泛舟的遊戲。

主角為台大人駕著船，在舟山河上行駛，目標是躲避各種依不同速度撞向船的障礙物。每個障礙物的損害都不一樣，速度也有差異。若撞到障礙物，左上角的GPA以及GPA血條會依照受損程度改變，若GPA<0，則遊戲結束，會詢問是不是要重新玩 ; 反之若成功撐到關卡開始後30秒(總遊戲時長150秒)，則終點線會出現，只要船撞到終點線，遊戲即結束，並顯示該關所得最後之GPA。

**系統設計與演算法**

本遊戲使用的Python套件包含：pygame, time, random, logging, statistics, math等等，主要使用pygame模組搭配其它套件做出功能。

本遊戲所有顯示在螢幕的物件都是由我們親自手繪而成。

由於Pygame的顯示方式是以很快的速度更新螢幕，並透過每次更新螢幕時x,y的座標變換來顯示物件，因此會造成物件在螢幕上移動的錯覺。

首先，主要是設計船隻，在press keydown時，船隻的移動速度為15px，也就是只要啟動上下左右鍵，就會在下次更新螢幕時，距離原本的點15px。

再來是物件的設計，我們利用線上的pixel 繪圖網站，繪製障礙物物件，並使用class來統一設置物件的移動速率，大小，初始位置，撞到時受到損傷的程度等等。

值得一提的是，為了讓船隻可以顯現出GIF的變換效果，我們微調了船隻的pixel圖，並讓螢幕每次更新5幅畫面，就會顯示另外一張圖片，並一次顯示，造成類似GIF的效果。而舟山「河」的河流底圖也用同樣的道理，讓三種不同的藍色以隨機 (random)的方式呈現在螢幕上。

碰撞的部分，我們設定成只要船隻的y座標落在障礙物的y座標之間，且x座標介於障礙物的y座標之間，那就視為有碰撞到。只要有碰撞到到情行產生，那船隻就會自動開始閃爍，螢幕每更換10次閃爍一次，並持續3秒的時間。所有的以上功能均有經過class的模組化，方便我們做不同物件的操控與修改。

**分工方式**

第二關的遊戲「舟山河」由兩位組員：李杰臨、李季陽 完成。

李杰臨：

將code模組化 (變成class 與function)

設定右上角時間的運行。

設定左上角GPA的控制。

設定船隻與障礙物碰撞後的閃爍反應。

定義船隻與障礙物的碰撞事件發生與否。

設定障礙物功能。

繪製物件。

將第一關到第三關的code 做結合與模組化。(merging)

李季陽：

將code模組化 (變成class與function)

設定左上角GPA圖示與血條的控制。

設計螢幕上方底圖變換。

設定底圖河流顏色閃爍的問題。

設定船隻GIF變換的方式。

設定障礙物功能。

繪製物件。

將第一關到第三關的code 做結合與模組化。(merging)

**心得**

李杰臨：

這次實作pygame實在遇到許多出乎意料之外的困難，是第一次在完全沒接觸過pygame的情況下，從無到有產出一個python的遊戲，也與季陽一同體驗在一週的時間內完成coding接力賽的挑戰。常常是季陽開工完後，換我做，然後push到github上進行版本控制，並在交班時，還常常打電話解釋彼此的code在幹嘛，是體驗成為一位programmer會遇到的困難的難得體驗。過程中，也印證了一句話：「沒有寫不出來的code」。看得只是邏輯與心力付出的差別而已。最麻煩的當然還是結合三關的code，足足花費了我們一整天的時間，也體會到版控的重要性。

李季陽：

這次實作Pygame對我來說是一個非常有意思的體驗，從小到大我一直有個夢想是成為一位遊戲設計師，透過這次的實作，算是讓我初步的了解遊戲的運作原理(不知道其他引擎的原理是否一樣就是了)。從遊戲企劃，遊戲功能設計，遊戲美術設計，debug，全部一手包辦，真的不是一件容易的事情，光是這麼一哦簡單的遊戲其實就忙得我們不可開交，更何況是那些跨時代的巨作呢?

最難的部分其實是整合其他人的code與組員之間的版本控制，到後期，三不五時就會merge一下，拖累整個專案的作業進度，也讓我們咧解版本控制的重要性，以及如何適當處理conflict。我對於這次的成果還是感到非常滿意的，謝謝這堂課給我們這個機會來自己設計遊戲。