



計算機程式設計期末專題

台灣編年史

Taiwan Chronicle

地質四 B02204028 魏孫正

地質四 B02204034 江侑倫

地質三 B03204005 蔡承軒

一、 題目之動機

本學期修習老師的程式設計，獲益良多，學習到了程式設計的基礎，望能利用這些基礎，寫出完整的遊戲，供大家使用。另一方面，本學期因為修習台科大之遊戲心理學課程，而設計了教育性的桌遊，望能利用此機會，將此桌遊程式化，讓更多人能夠使用，藉以修正遊戲規則，並加強教育性。這個桌遊主要的教育目的在讓更多的小朋友能夠了解並認識台灣這個島嶼上特有種動物的習性，以及在地質研究上所發現在地球不同時間所發生的各種事件，對環境及動物造成的影響等。

二、 構想解說與遊戲規則說明

我們分別寫了兩種程式，一是較簡單的主控台程式，二是結合 unity 所設計出來的遊戲。兩個版本的遊戲規則均相同，惟透過 unity 所設計的程式仍有一些未完成的部分，還望多多見諒。

這邊將針對遊戲規則進行進一步的說明：

1. 卡牌

本遊戲中的卡牌，大致可以分為下列四種：角色卡、商店卡、世界卡以及食物卡。

(1) 角色卡

代表各玩家的卡牌，遊戲開始時，各玩家可以從三位角色中選擇一張做為代表自己的角色，各個角色分別擁有不同的技能，藉此反映出牠們的習性。(角色的技能與選擇部分尚未完成，會在下個版本推出)

(2) 食物卡

能回復飽食度的卡片，飽食度相當於行動度，大部分的行動都需消耗飽食度玩家須在回合結束後，選擇食物並進行搶食，以獲得食物卡來回復飽食度。

(3)商店卡

玩家能夠在行動階段利用 DNA 進行購買的卡片，DNA 會在每回合開始時，根據玩家自身的數量回復。商店卡在購買後即加入玩家的手排並且可以馬上使用。

(4)世界卡

每回合開始時抽出的卡片，世界卡會對所有場上的玩家造成影響。

2. 特殊行動與名詞說明

(1)**搶食**：玩家能在這個階段使用基本卡及商店卡暫時強化自己，搶食的結果由玩家的速度及力量決定，速度快者獲勝，若速度相同時則力量大者獲勝。

(2)**飽食度**：上限為 6，歸 0 時判定死亡，可以透過食物卡來回復。

(3)**數量**：上限為 6，歸 0 時判定死亡，達上限時，維持至下一輪則判定勝利。

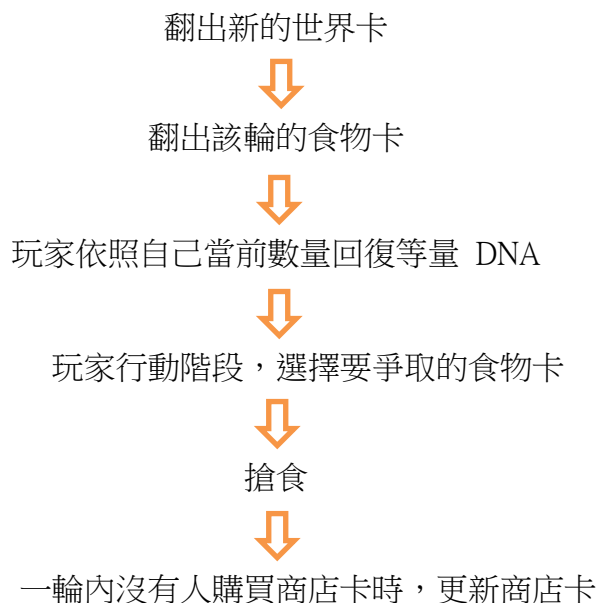
(4)**DNA**：購買商店卡所需的貨幣，每回合依據玩家數量補充，上限 10。

(5)**迴避**：攻擊判定失敗。

(6)**繁殖**：玩家可不經由卡片，消耗 3 飽食度以增加 1 數量。

(7)**攻擊**：消耗 1 飽食度進行攻擊，基本攻擊造成 1 點傷害，每回合限 1 次。

3.回合流程



4. 玩家行動階段

(1)玩家可在自己回合使用任何商店卡並消耗相對應的飽食度，並可在自己回合的任何時候購買商店卡並可馬上使用

(2)玩家可在回合中使用攻擊指令

(3)結束回合時需選擇欲進行搶食的食物

5.勝利條件

(1)玩家在他的回合及下次回合開始時數量皆在 6 者，判定其勝利

(2)只剩下一位玩家時，判定其勝利

三、程式測試規劃

(1)主控台方面利用 class Player 紀錄玩家資訊、提供找出手牌空格、裝備空格的方法

(2)利用 class Card 紀錄讀取進來的卡片，提供 use 方法讓卡片與玩家互動，

(3)利用 class World 紀錄世界卡資訊，主要是對各種食物的飽食度增減

(4)利用 class Food 紀錄食物卡資訊，紀錄食用該食物所需的能力值

(5)class AI 紀錄 AI 想過的事，如 bool GoShopping 紀錄 AI 是否該呼叫 Shopping 方法來計算該買甚麼，如在 Shopping 裡發現沒有想買的，或是有很想買的卻買不起的卡片則設為 false，AI 此回合便不會再去買卡片，Use 方法與 Shopping 方法類似，提供 reset 方法用來在下個回合開始時重置 AI，提 AIchoise 方法讓 AI 決定要買東西、使用卡片或攻擊

(6)利用 class Deck 紀錄場上卡片的索引值，以用來進入 class Card 取得卡片資訊，提供 getLastspace 方法，獲取 Deck 中索引值陣列的空格，以用來加入新卡片進入該牌堆

(7)資料讀取的部分，以功能牌為例，首先為每種功能牌分別建立 txt 文字檔，並分行寫下該種功能牌的名稱、數量、耗費等資訊，然後存在同一個資料夾 card 中；建立一個長度與 card 裡文字檔數一樣的字串陣列，以檔名指定對應位置的元素字串；建立一個 Card 類別的陣列 card，先為每個元素初始化，然後使用 StreamReader file 以文字檔裡的資訊指定 card 的特定元素的每個變數，重複以指定 card 的每個元素的每個變數。

(8)遊戲初始化的部分，首先需要建立功能牌的遊戲總牌庫 Card 陣列，其每一個元素即代表一張牌，往後卡牌的移動等等，便只需以整數作處理，再當作索引到遊戲總牌庫尋找需要的資料。

(9)接著，需要將食物牌與功能牌洗牌，但是遊戲總牌庫的順序不能變動，故暫時建立一個長度與總牌庫相同的連續非負整數陣列起始牌庫，洗好牌後，將前二十張牌放入玩家手牌，接著的八張放入商店，再把剩下的功能牌放進商店牌庫。

(10)另外，玩家與 AI 的各項數值則依據規則作初始化。

(11)每一輪利用 do - while 及 leftOne 函式檢查是否剩下一位玩家，是則結束遊戲，否 則進入下一輪，每輪開始時抽出並顯示世界卡及食物卡，計算每個食物與世界卡裡 debuff 們的互動，顯示實際吃到時回復的飽食度

(12)回合中玩家選擇要做的事並叫出對應的函式 use、buy 及 attack，利用 while 在選擇離開之前不結束回合

(13)回合結束時 currentPlayer 加一，並進入下個回合，以 AI.AIchoise 方法選擇叫出 use、buy 或 attack，在各函式裡呼叫 AI 裡的 Using、Shopping 或 Attacking。玩家與 AI 皆能直接使用 use、buy 及 attack

(14)每輪結束呼叫 eat 函式計算誰搶到甚麼食物，並回復飽食度

(15)Unity 方面，將 console 版回合開始前的程式丟進 readfile 及 initial 函式，以變數 ShopPos、HandPos、FoodPos 紀錄卡片該在的位置

(16)只要在指定的資料夾加入 txt 檔及圖檔便能加入新的卡片，以卡片的 name 去找到對應的 png

(17)分別建立手牌、商店及食物卡的 Sample，以用來建立實際物件並丟到指定的位置，每個物件都是按鈕，按了之後會對應到指定的用來顯示大圖的物件，在該物件上顯示該按鈕的圖，如按的是手牌或商店則顯示使用或購買選項，將卡片在手牌內或商店內的位置傳入用來顯示之物件的 int Index，按下購買或使用按鈕後將 Index 傳入主程式並呼叫對應的 buy 及 use

(18)AI 做的事採用 Doge Message System(暫定)顯示，因其顯示結果很像 Doge 圖得名

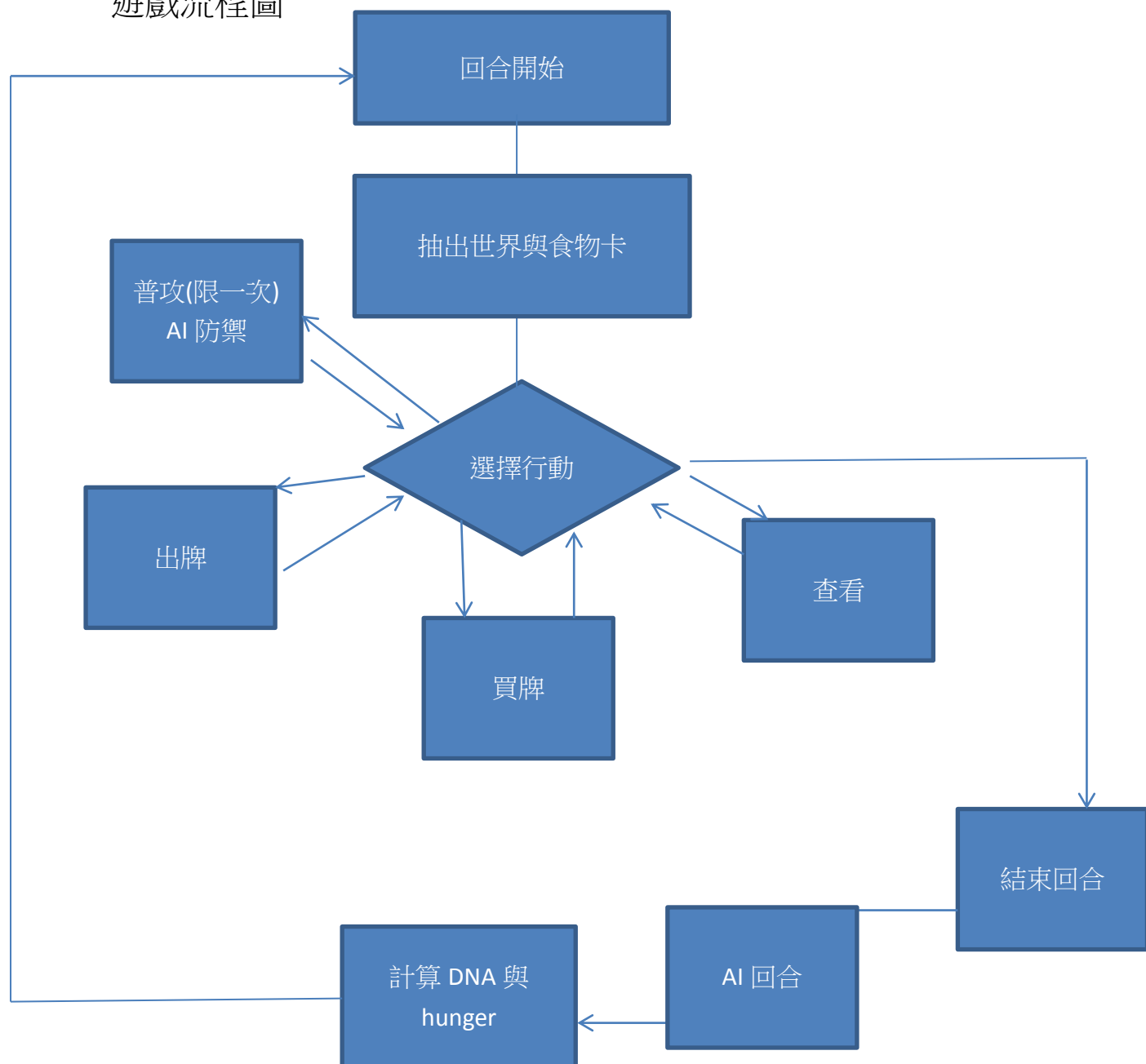


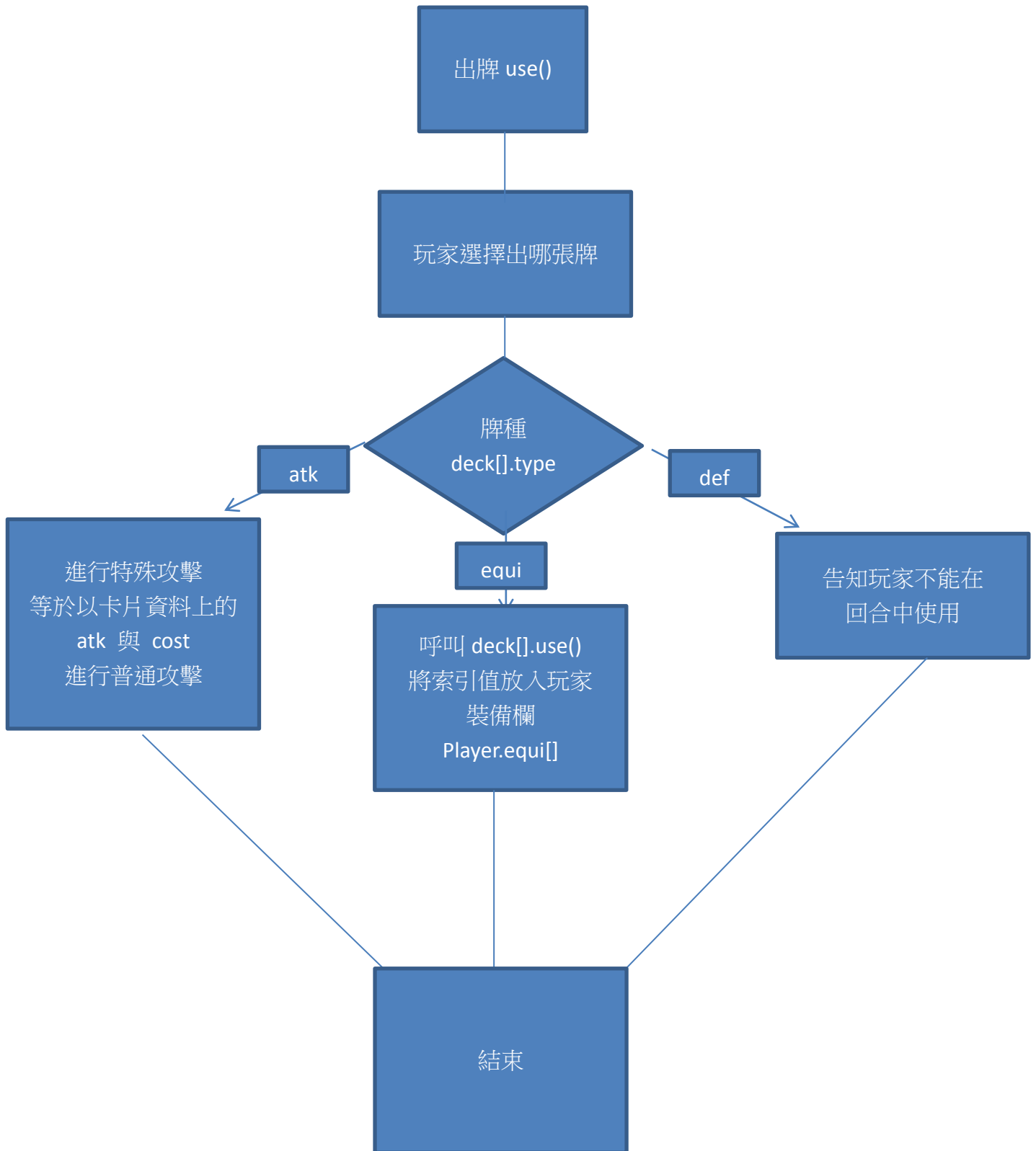
(圖)Doge

(19)使用 showMessage 函式建立一個 Text 物件並放到畫面上隨機一個位置，Much cool，Such interesting

四、 流程圖

遊戲流程圖





*deck[]是牌的資料庫，Deck 類別是用來紀錄索引值以取出裡面的資料

*普通攻擊消耗 1 飽食度最多造成 1 傷害

五、程式列表

詳見專案檔

六、程式測試執行結果

請參考 youtube 影片

七、參考文獻

鄭士康, 以 C#學習程式設計基礎:使用 Visual Studio 2012 Express for Windows Desktop, 2013.

MSDN, <https://msdn.microsoft.com/zh-tw/dn308572.aspx>

Unity, <https://unity3d.com/>

八、學習心得

魏孫正

經過這學期老師的教導，學習到程式設計的技巧，並且透過各個作業的練習，在學期的最後寫出屬於自己的專題，完成的瞬間是令人非常喜悅的一件事情。在期末的專題中嘗試使用 GitHub 以及 Unity 這兩種程式來協助開發，然而過程並不如想像中容易，GitHub 的同步有時候會把修改的東西吃掉，Unity 在卡片遊戲的製作上，找不太到教學，所以很多東西都要自己嘗試，加上遊戲寫出新東西後的 debug 方面，亦花了不小的時間。

江份倫

設計出一個桌遊並實際自己玩它，是我從沒想過的，把他寫成程式更是沒想過，寫程式的樂趣在於將構想實踐的快感，寫這個程式讓我知道甚麼是廢寢忘食，寫到後來連作夢都會夢到程式碼，很有趣，但有點恐怖

在把程式 Unity 化的時候，我深深感受到資訊的價值，網路上許多免費教學都不知道初學者真正要的是甚麼，很多我想使用的方法都沒有實際例子，害我都要試很久，要賣錢的教學應該會寫的比較好吧，我猜。相較之下，老師的講義真的很清楚的演示每個操作細節，讓我能快速理解每個概念。

蔡承軒

這次參與專案的製作是個很特別的經驗，讓我了解到多人合作編程真的是一件不簡單的事，最大的難點就是資訊不一定能被完整地傳達或理解，而且若沒有事前縝密地討論計畫，經常要花大量的時間來理解同伴撰寫的程式碼，使工作效率大

幅下降；其次就是大家常常忘記用 **Git** 上傳自己最新的變更，導致經常出現大家為了能順利偵錯而各自修改同一個錯誤，耗費相當多的時間。