

# 【愛吃鬼 XX】

## 接東西小遊戲

By : GROUP\_4

【408272643】 【鄭衣彤】

【409570224】 【林虹樺】

【409570236】 【吳思嫻】

【409570298】 【王殿馨】

Introduction to Computer Science

Fall 2020

Date Submitted : January 17, 2021

# 目錄

一、	遊戲製作動機及參考.....	2
二、	遊戲介紹.....	3
三、	遊戲規則.....	3
四、	研究內容	
	(1). 介面設計.....	5
	(2). 程式碼展現.....	8
	倒數計時、計分表、頁面跳轉、動物左右移動、物件隨機產生、物件觸及某物後消失、物件消失後加減分	
五、	遊戲測試.....	15
六、	專案成本.....	18
七、	遭遇困難及解決方式.....	19
八、	團隊分工.....	21
九、	心得.....	22

## 一、遊戲製作動機及參考：

才剛初學程式語言的我們，還未有能力做出更艱深更複雜的程式設計，於是我們想先從簡單的小遊戲著手，藉由實作來提升自我能力及經驗。

原先我們在選擇要做甚麼小遊戲程式時，想從簡單且大家都玩過的那種遊戲為參考，最廣為人知的小遊戲莫非就是大家小時候玩的網頁遊戲，類似「遊戲天堂」(遊戲網頁)中尋找，找到了幾個我們認為適合挑戰的遊戲，我們挑出「電眼美女」、「打磚塊」、「小貓跳繩」、「吃東西變大」這四款遊戲做出投票，由於我們都是吃貨(喜歡吃東西的人)，最後由「吃東西變大」為遊戲主題發想。



## 二、遊戲介紹：世界觀描述

在彩虹世界的歐虧王國，裡面住了超級多可愛的小動物們，他們都過著幸福快樂且不斷吃東西的日子，每天都在「吃飽睡、睡飽吃」。其中，王國裡有五位很愛吃的子民，然後他們常常在爭執誰吃得最多。

目標是在關卡 20 秒中，幫助我們所選擇的子民，吃到最多「能吃」的食物，遊戲同時也會出現「不能吃」的食物，所以要請遊玩者小心注意「接住或閃躲」食物，得到高分贏得歐虧王國裡的愛吃鬼寶座，變成肥肚肚左衛門！

## 三、遊戲規則：加扣分機制

### (1). 角色：狗

可以吃的(加分)：狗骨頭+5、肉+5、巨大狗骨頭+10

不可以吃的(扣分)：罐頭-5、巧克力-5、炸彈-10

### (2). 角色：貓

可以吃的(加分)：小魚乾+5、蝦+5、巨大小魚乾+10

不可以吃的(扣分)：橘子-5、辣椒-5、炸彈-10

(3). 角 色：熊貓

可以吃的(加分)：竹子+5、窩頭+5、巨大竹子+10

不可以吃的(扣分)：鞋子-5、鬼牌-5、炸彈-10

(4). 角 色：兔子

可以吃的(加分)：胡蘿蔔+5、牧草+5、巨大胡蘿蔔+10

不可以吃的(扣分)：肉-5、辣椒-5、炸彈-10

(5). 角 色：豬

可以吃的(加分)：飼料+5、玉米+5、蘋果+5、巨大蘋果+10

不可以吃的(扣分)：汽油-5、塑膠袋-5、炸彈-10

#### 四、研究內容：

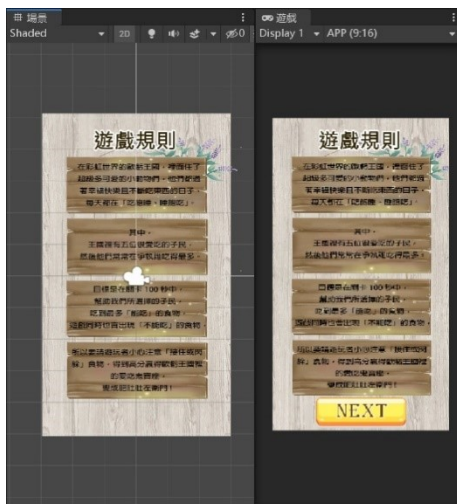
##### (1). 介面設計



##### 【第一頁】首頁

我們在首頁設計放入 UI 按鈕鍵，使其可以跳轉至下一頁。

第一頁的功能是讓使用者清楚知道我們的遊戲標題，還有從遊戲封面可以看出來我們的遊戲風格以及跟食物相關的主題。



##### 【第二頁】遊戲規則

我們在第二頁設計放入 UI 按鈕鍵，使其可以跳轉至下一頁。

第二頁的功能是讓使用者了解遊戲背景，進而知道遊戲運作故事及規則。



##### 【第三頁】選擇角色

我們在第三頁設計放入 5 個 UI 按鈕鍵，使其可以跳轉至下一頁，但此 UI 鍵跟前幾頁的順序不一樣，角色的按鈕會對應不同的角色的第四頁。

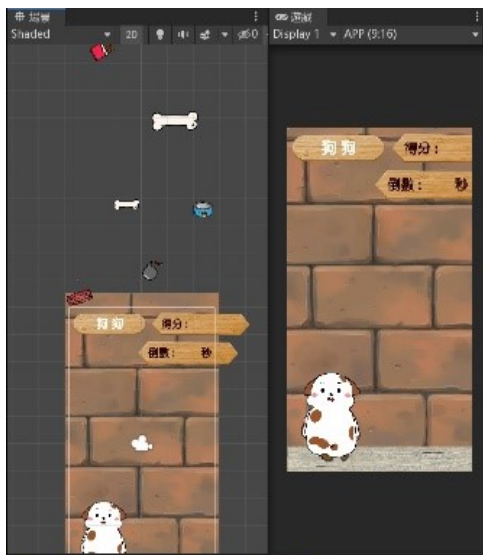
第三頁的功能是讓使用者選擇角色，這會影響後來遊戲不同介面的轉換。



#### [第四頁] 遊戲食物說明

我們在第四頁設計放入 UI 按鈕鍵，使其可以跳轉至下一頁也就是遊戲開始。

第四頁的功能是讓使用者了解所選之角色對不同食物是否可以吃的規則，這會影響後來遊戲進行裡的加分、扣分計算。



#### [第五頁] 遊戲進行中介面

我們在第五頁設計放入 UI 鍵(點擊結束)，使其可以跳轉至下一頁，遊戲結束頁面。

在第五頁裡設計物件方塊，並設計其相關屬性(鋼體、2d collider(碰撞器))，加入程式碼(碰到角色加分扣分、跑出介面被銷毀、隨機產生物件、由上往下落下速度)遊戲結束頁面。

在第五頁裡設計介面中，還有計分、計時的相關程式碼，利用程式碼與 text 物件的交互配合，使其呈現在頁面中。



### [第六頁] 遊戲結束

我們在第二頁設計放入兩個 UI 按鈕鍵，使其可以跳轉至角色選單，再玩一次遊戲，或者退出遊戲。

第三頁的功能是讓使用者選擇是否要讓遊戲再執行一遍，或者退出遊戲畫面。



## (2). 程式碼展現

### (A) 倒數計時：

```
using System.Collections;↵
using System.Collections.Generic;↵
using UnityEngine;↵
using UnityEngine.UI;↵
|↵
↵
↵
public class CountdownTimer : MonoBehaviour↵
{↵
    float currentTime = 0f;↵
    float startingTime = 20f;↵
    ↵
    [SerializeField] Text countdownText;↵
    ↵
    void Start()↵
    {↵
        currentTime = startingTime;↵
    }↵
    void Update()↵
    {↵
        currentTime -= 1 * Time.deltaTime;↵
        countdownText.text =
currentTime.ToString("0");↵
    }↵
    ↵
    if(currentTime <= 0)↵
    {↵
        currentTime = 0;↵
    }↵
    }↵
}
```

(B) 計分表：

```
using System.Collections;↵
using System.Collections.Generic;↵
using UnityEngine;↵
using UnityEngine.UI;↵
↵
↵
↵
public class Scoregain : MonoBehaviour↵
{↵
    public Text scoretext;↵
    public int score;↵
    ↵
    public void AddScore(int value)↵
    {↵
        score += value;↵
        UpDateScore();↵
    }↵
    void UpDateScore()↵
    {↵
        scoretext.text = "" + score;↵
    }↵
}↵
```

(C) 頁面跳轉：

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class menu : MonoBehaviour
{
    public void Gamestart()
    {
        SceneManager.LoadScene("2");
    }
    public void Gamemenu()
    {
        SceneManager.LoadScene("3");
    }
    public void DogGame()
    {
        SceneManager.LoadScene("4-1");
    }
    public void CatGame()
    {
        SceneManager.LoadScene("4-2");
    }
    public void RabbitGame()
    {
        SceneManager.LoadScene("4-3");
    }
    public void PandaGame()
    {
        SceneManager.LoadScene("4-4");
    }
    public void PigGame()
    {
        SceneManager.LoadScene("4-5");
    }
    public void Playdog()
    {
        SceneManager.LoadScene("5-1");
    }
    public void Playcat()
    {
        SceneManager.LoadScene("5-2");
    }
    public void Playrabbit()
    {
        SceneManager.LoadScene("5-3");
    }
    public void Playpanda()
    {
        SceneManager.LoadScene("5-4");
    }
    public void Playpig()
    {
        SceneManager.LoadScene("5-5");
    }
    public void QuitGame()
    {
        Application.Quit();
    }
    public void Menu()
    {
        SceneManager.LoadScene("Menu");
    }
}
```

(D) 動物左右移動：

```
using System.Collections;↵
using System.Collections.Generic;↵
using UnityEngine;↵
↵
public class PlayerControll : MonoBehaviour↵
{↵
    ↵
    ↵
    ↵
    ↵
    ↵
    void Start()↵
    {↵
        ↵
    }↵
    ↵
    ↵
    void Update()↵
    {↵
        ↵
        ↵
        if(Input.GetKey(KeyCode.LeftArrow))↵
        {↵
            transform.localPosition += new Vector3(-10,
20, 0) * Time.deltaTime;↵
        }↵
        if(Input.GetKey(KeyCode.RightArrow))↵
        {↵
            transform.localPosition += new Vector3(10,
20, 0) * Time.deltaTime;↵
        }↵
        ↵
        ↵
    }↵
}↵
↵
```

(E) 物件隨機產生：

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class GameFunction : MonoBehaviour
{
    public GameObject Food1;
    public GameObject Food2;
    public GameObject Food3;
    public GameObject Food4;
    public GameObject Food5;
    public GameObject Food6;
    public float time;

    void Start()
    {
    }

    void Update()
    {
        time += Time.deltaTime;
        if (time > 20f)
        {
            Vector3 pos = new
            Vector3(Random.Range(-2.5f, 2.5f), 5.4f, -0.1f);
            Instantiate(Food1, pos,
            transform.rotation);
            time = 0f;
        }

        time += Time.deltaTime;
        if (time > 20f)
        {
            Vector3 pos = new
            Vector3(Random.Range(-2.5f, 2.5f), 5.4f, -0.1f);
            Instantiate(Food2, pos,
            transform.rotation);
            time = 0f;
        }
    }
}
```

```
time += Time.deltaTime;
if (time > 20f)
{
    Vector3 pos = new
    Vector3(Random.Range(-2.5f, 2.5f), 5.4f, -0.1f);
    Instantiate(Food3, pos,
    transform.rotation);
    time = 0f;
}

time += Time.deltaTime;
if (time > 20f)
{
    Vector3 pos = new
    Vector3(Random.Range(-2.5f, 2.5f), 5.4f, -0.1f);
    Instantiate(Food4, pos,
    transform.rotation);
    time = 0f;
}

time += Time.deltaTime;
if (time > 20f)
{
    Vector3 pos = new
    Vector3(Random.Range(-2.5f, 2.5f), 5.4f, -0.1f);
    Instantiate(Food5, pos,
    transform.rotation);
    time = 0f;
}

time += Time.deltaTime;
if (time > 30f)
{
    Vector3 pos = new
    Vector3(Random.Range(-2.5f, 2.5f), 5.4f, -0.1f);
    Instantiate(Food6, pos,
    transform.rotation);
    time = 0f;
}
}
```

(F) 物件觸及某物後消失：

```
using System.Collections;↵
using System.Collections.Generic;↵
using UnityEngine;↵
using UnityEngine.UI;↵
↵
public class Foodgone5 : MonoBehaviour↵
{↵
    ↵
    ↵
    ↵
    ↵
    private void OnCollisionEnter2D(Collision2D
collision)↵
    {↵
        ↵
        ↵
        if(collision.gameObject.name == "物件碰到
的東西" )↵
        {↵
            FindObjectOfType<Scoregain>().AddScore(要加減幾分);↵
            Destroy(gameObject);↵
        }↵
    }↵
}↵
```

(G) 物件消失後加減分：

```
using System.Collections;↵
using System.Collections.Generic;↵
using UnityEngine;↵
using UnityEngine.UI;↵
↵
public class Foodgone5 : MonoBehaviour↵
{↵
    ↵
    ↵
    ↵
    ↵
    private void OnCollisionEnter2D(Collision2D
collision)↵
    {↵
        ↵
        ↵
        if(collision.gameObject.name == "物件碰到
的東西" )↵
        {↵
            FindObjectOfType<Scoregain>().AddScore(要加減幾分);↵
            Destroy(gameObject);↵
        }↵
    }↵
}↵
↵
```

## 五、遊戲測試：遊戲流程



第一頁按下【START】後跳轉至第二頁。

第二頁為遊戲世界觀介紹，閱讀完後按下【NEXT】跳轉至第三頁。

第三頁是選擇想要的動物，點選完後跳至相應的頁面。





第四頁是介紹所有動物能吃和不能吃的東西，選擇完後按下【GO】即可開始遊戲。



第五頁即為遊戲畫面，遊戲時間為20秒。左右拖曳動物即可接東西。

時間結束後跳轉至第六頁。



第六頁按下左邊的【EXIT】即為結束遊戲。

按下右邊的【AGAIN】會跳回第一頁。

## 六、專案成本：

前期籌備討論時間：路易莎4小時，200／小時，總共3200。

餐費150／人，總共600。

設計／繪圖：約6小時，200小時，總共4800。

介面設計、物件控制：共15小時，200／小時，總共12000。

程式碼編寫：共30小時，200／小時，總共24000。

最後合併檔案：摩斯漢堡5小時，200／小時，總共4000。

餐費150／人，總共600。

（以上工時皆＊4人）

總計60小時＊200＊4人＋餐費150＊8＝49200元

## 七、遭遇困難及解決方式：

- 不知道該使用哪種程式/軟體

其實我們四個都不太懂怎麼寫程式該用哪個軟體寫會比較好，不知道該用甚麼軟體去呈現，所以當初去就帶著這個問題去詢問老師，當時老師推薦使用 unity 來做遊戲會比較好。最後我們使用了兩個軟體，unity 以及 visual studio。

UNITY 的優點也很多，像是他不用錢，我們不必額外去找破解版或付錢給它就可以從官網上下載直接使用而且它有很多遊戲模式可以去設定選擇 2D 或 3D 模式也相對操作方便；而 visual studio 是用來寫 C# 腳本的工具，當我們在 UNITY 建立一個程式碼文件並且要打開時，是透過這個軟體去編譯程式碼的。

- 程式下載

下載 unity 花了我們很多時間，因為下載完還要啟動授權，而且從官方下載完後還要再下載安裝包，有時候還要設定東西就會導致下載失敗；我們的解決方法就是上網找很多人的安裝 unity 下載教學，最後大家都成功開啟 unity。

- 小組討論

我們這組對於製作遊戲都沒有概念，所以最後完成品與當初 proposal 遊戲設計有些出入，再改變及尋找遊戲碼的過程中，我們花了很多時間，但是要全組聚集一起討論時，有些人因打工、重要考試、通識課以及其他私人事情，有時不能順利喬出大家都可以討論的時間；解決方法是我們在設定討論時間時，盡量避開大家不行的時間，若真的有人不能來，我們就使用 Line 的線上通話功能，或我們討論完告訴沒來的成員今日的討論內容、進度，讓大家都可以就算不再討論當下，也能順利接收到討論內容，執行自己被分配後的任務。

- 遊戲程式碼執行錯誤

下載完 unity 但我們又遇到第二個困難，因為 Unity 使用的語言是 C#，我們都沒有人學過，大家僅具備簡單的 java 寫程式技能，所以對於全組來說要用它寫程式無疑是一項艱難的事。解決方案是我們遇到需要的程式碼就會上網搜尋找查，雖然花費時間很久，而且很多都是要打成英文才搜尋的到，但是我們都沒有放棄，把能用的程式碼留下慢慢修改成自己需要的，最後順利成功完成了此遊戲。

## 八、團隊分工：

這次的專案幾乎都是大家同時進行討，並做出來的，由於團體作業需要有人負責監督進度，及及時提出問題，故我們在設計、程式碼以及簡報、書面的部分都選出了一位負責的組員來做掌控進度的工作。

- 遊戲構思、架構：衣彤、虹樺、思嫻、殿馨
- 遊戲介面設計：衣彤、思嫻
- 遊戲程式碼：虹樺、殿馨
- 合併檔案：思嫻、虹樺
- 簡報、書面：衣彤

## 九、心得：

### (1). 【408272643】【鄭衣彤】

我們大家在這次的專題之前，都沒有接觸過Unity 和 c# 的語法，所以這專案對我們的挑戰真的非常大。我一開始為了學習 Unity 的使用方式，看了超過20小時的影片來了解每一個位置的功用。經過這次的專題作業，我學習到如何使用Unity，以及一些c#的程式編寫技巧。除了這些之外，我非常高興能和這三位學妹同組一起學習，他們對於程式碼編寫非常有挑戰的慾望，因此他們想做的功能算是滿多的。我們算是先分別搜尋資料，再一起做統整，雖然一開始挫折真的非常多，但大家還是共同克服了難題，完成了一份非常棒的專題作業！在看到程式能夠真正跑起來時，真的非常的開心，也非常感動！希望之後有機會的話我能將功能做得更加完善，且操作介面看起來更美觀！

(2). 【409570224】【林虹樺】

從這次的專題中，我學到了很多很有用的知識，像是 unity 的操作還有一些 c#簡單的語法，除此之外，也學習如何小組分工合作，讓專題更有效率的完成。在小組中，我負責找主要的程式碼，由於對 c#完全零基礎，所以在出現 bug 的時候非常痛苦，但也從中學習要保持耐心解決難題，並且享受問題漸漸被解決的成就感。從一開始完全不懂 c#，也完全不知道 unity 要如何操作，我藉由看網路上各式各樣的教學影片和文章，從最基礎的開始做起，找到類似的程式碼再改成自己需要的，最後在自己慢慢逐字理解吸收。很開心能有這個合作製作小遊戲的經驗，以前學程式因為都沒有實際應用過，所以比較沒有成就感，完成小遊戲後才了解到程式碼的重要性，也更想要好好學習寫程式了！希望能藉由這短短幾年的時間讓寫程式的能力越來越精進，也謝謝組員們非常給力的合作，才能有今天這麼棒的成果。



(3). 【409570236】【吳思嫻】

在這次的專題中，我學到了很多技能、關於時間的分配、小組分工。從未接觸過製作遊戲，一開始完全沒有概念，覺得小組因為會有每周進度在所以肯定不會拖到最後一周才完成遊戲設計。結果卻出乎意料，因為很多程式碼不是找到就好，需要套進去看看有沒有成功，還有介面設計完全都要憑空想像，可以說我們這組的遊戲都是完全用想像中的樣子去實現的。至於畫遊戲物件，並沒有我想像中的簡單，在遊戲製作進行時，我們加入了很多個人的想法，有些時候要多出一個 button 或物件，就需要再畫一個 png 圖檔，但是我覺得在製作遊戲程式和繪製遊戲物件的過程，我更熟悉兩者的使用方法，這對未來必定有幫助。

時間分配在製作專題也是很重要的一環，我要謝謝小組中的所有成員，在每次開會後大家都有盡心把每一個任務都執行好，沒有都不想做事的人，雖然製作時很心累一直失敗、一直畫物件，但是有這麼努力配合的組員們，我認為技能學到關於遊戲製作技巧，還能配合到一群負責任有擔當，且當有人有難時全組都會一起互相幫助，真的很幸福。

(4). 【409570298】【王殿馨】

這次的專題中，一開始選定了「接東西」的遊戲後，開始構想整  
的遊戲的架構。一開始都想得很美好，有很多角色、背景，掉落下來的  
東西還要是隨機的。在專題中，我和虹樺一起負責的部分是遊戲的  
程式（不包括跳轉及按鍵）。在進入這個科系之前，我從來沒有接觸  
過任何的程式語言。在上榜到現在，也只有些觸過 Python 和 Java 而  
已。但是我們選用的是 Unity，他所使用的程式語言是 C#。為了進行  
順利，一開始先查了很多關於 C# 的教學。在後來開始一個個去查我們  
所需要的程式，也發現需要更多教學時，因為中文的資料實在太少，  
只好去看全英文的教學影片。也讓我知道了，如果未來需要更多的知識  
量，英文是不可少的條件。很開心最後專題有順利交出，雖然還有許  
多不管是畫面或是程式需要修改的地方。但是看到可以玩的時候，真  
的覺得超開心。