

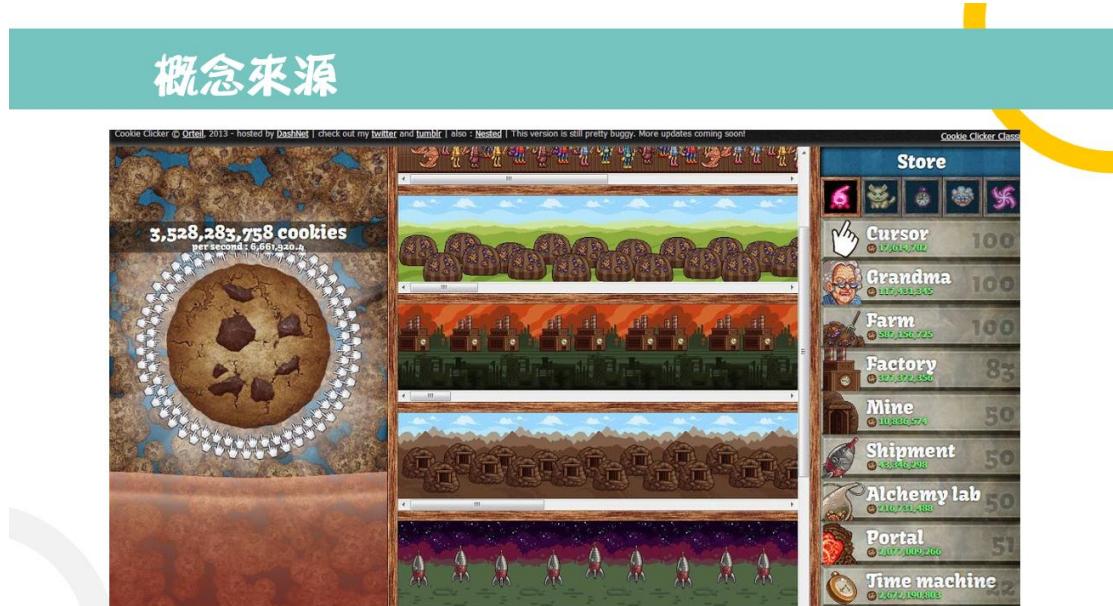
計算機程式設計

主題 放置遊戲

機械一 朱本毅

一、動機：

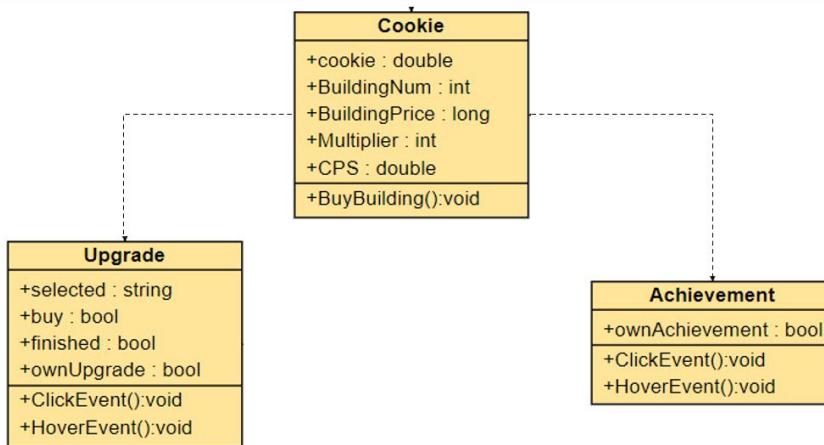
本次作業，內容主要是利用 Windows Form 的架構，寫出一個點擊 + 放置類型的遊戲。這是一種常見的遊戲類型，而我參考的原作：Cookie Clicker，在我小學的時候就有了，至今為止的十年中陸續有多次的更新，具備有非常豐富的系統及架構。先聲明，我只參考了其遊戲概念、特色（原作很多地方可以看出作者的幽默）、以及建築價格的公式，其他圖片（無商業利用的版權問題）、文字幾乎都是自己想的 😊。



二、程式架構與規劃:

本次的作業，我著重於三個方面：還原基本功能、升級項目、成就。第一個首要目標就是完成基本的架構。我用到的類別如下：

類別圖



Cookie:本程式所有涉及餅乾數量的部分，我就都放在這個類別裡了。

BuildingNum 代表建築的數量

BuildingPrice 是建築價格，遵循以下公式： $\text{Base Cost} * \text{BuildingNum}^{1.15}$

Multiplier 分為建築物的乘數以及全域乘數(Global Multiplier)

CPS 是「每秒獲得餅乾數」的縮寫，這和各種建築數量、升級項目的購買狀況、以及技能有關

我用的公式是：各建築 $\text{Base CPS} * \text{BuildingNum} * \text{Multiplier}$ 加總

BuyBuilding()是在點下購買按鈕時，判斷餅乾是否足夠並調整參數的函式

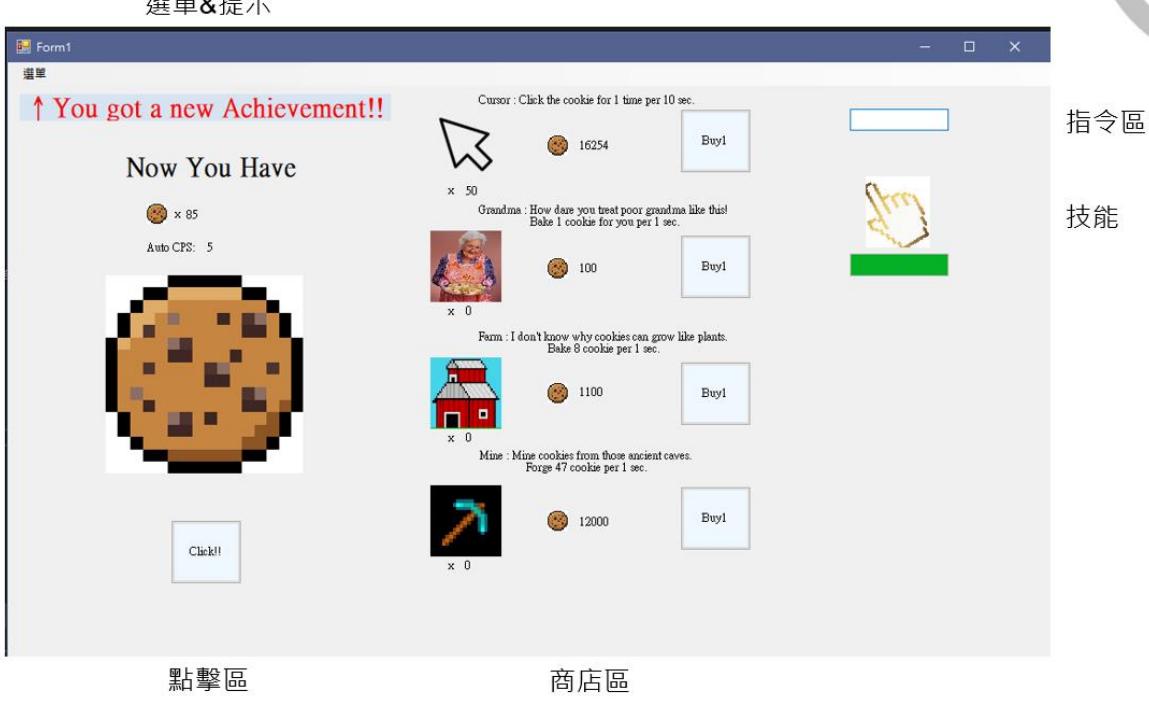
Upgrade:升級頁面，主要包含一些布林變數和事件，用來判斷我的購買狀況

Achievement:成就頁面，比較沒甚麼特別的

PS.

放置類型的遊戲，他的金錢(在本作中是餅乾)，往往都是天文數字，必須額外寫一套額外的進位系統，但我想做的只是小品遊戲，因此設定了一個上限以及明確的遊戲目標，就是達成所有成就。

三、主介面：



點擊區：明確的畫面，原作是直接點擊餅乾，並且會有動態效果以及音效。但我卻選擇用按鈕，因為這是 Form，所以.....只要你先按一次 Click，變成一個選取此按鈕的狀態，這時按住 Enter 就會自動一直點擊了(這麼貼心的連點功能^v^)。

餅乾的數量涉及一個函式 Update()

Update()

目的：

1. 每秒更新餅乾總數(autoCPS=每種BuildingNum*BuildingCPS*Multiplier)

```
CookieNum.Text = Math.Round(cookie.cookie).ToString();  
private void Timer_Tick(object sender, EventArgs e)  
{  
    cookie.cookie += (double)(cookie.autoCPS);  
    Update();  
}
```

(當購買第一個建築時，啟動此timer)

2. 更新建築數量、價錢

價錢公式:Base Cost*BuildingNum^{1.15}

3. 更新成就

當解鎖新的成就時，顯示提示訊息並且撥放提示音效(點開成就頁面後移除通知)

商店區：購買各種自動生產餅乾的建築(這只是通稱，老奶奶當然不算建築)

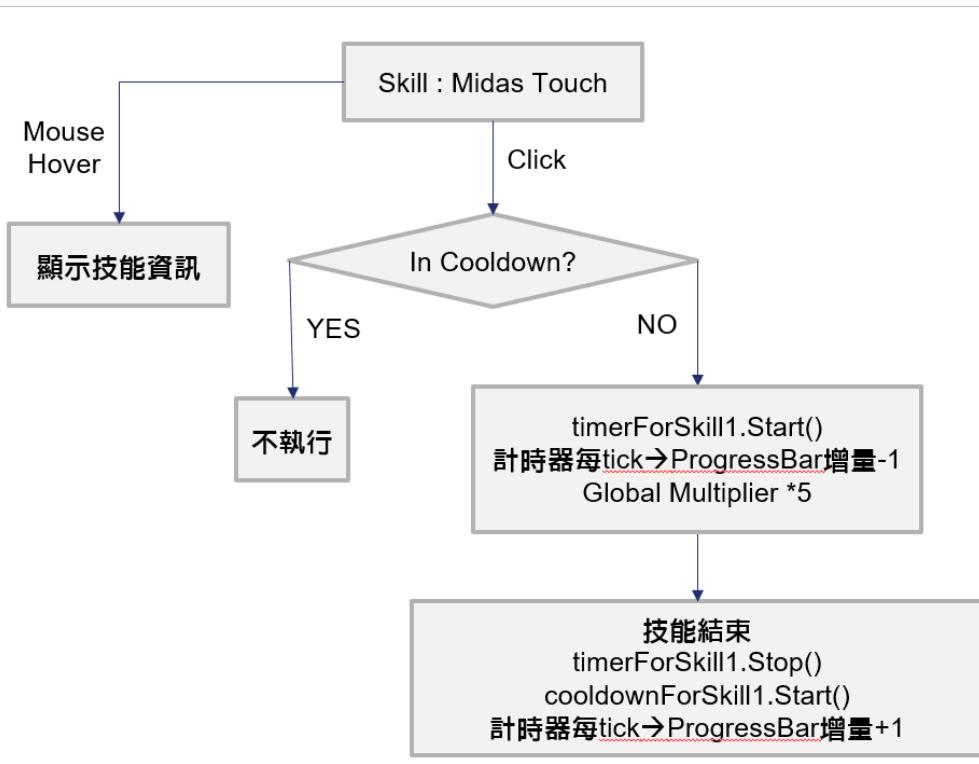
各種建築單位的上限是 100，買完後再按下 Buy 按紐即會顯示 Max 並使按紐

Enabled=False。同時，建築也和升級項目的解鎖條件有關。

指令區：輸入指令後按下 Enter 就會觸發，這是方便我自己在測試或 Debug 用的，想作弊的話可以自己翻程式碼 😞。

技能：原作中並沒有技能，但每隔一段時間就會有金餅乾，點到就會直接獲得一定數量的餅乾或是短時間內產量倍增。

Midas Touch 在購買完 Cursor 的最後一個升級項目後解鎖，點下去即可在一分鐘內加速五倍餅乾產量，進度條會隨著技能的使用而減少，亦會隨著冷卻時間緩慢恢復，在視覺上的效果非常明確！



四、升級介面：



原作中的升級項目是一個重頭戲，效果十分多元，但我做的比較簡單化，著重在加快遊戲進程。

每一種升級項目在點選的同時就會進行選取，指定給 selected 字串，並顯示底下的說明文字，如果已擁有的話則顯示 You already have this.。

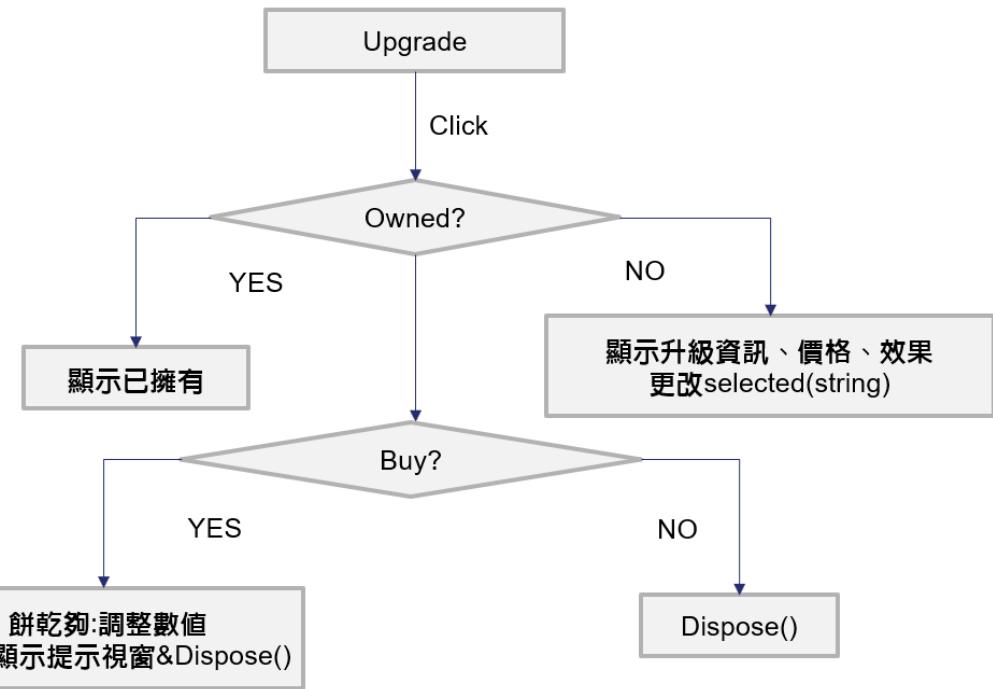
程式碼放在下一頁，每次開起購買頁面都會重製 buy、finished 這兩個變數，我用 while 迴圈和 switch 反覆進行判斷，如果我符合 1. 選定一個未購買的項目 2. 我的餅乾數量足夠，就會成功購買，若是餅乾數量不足或直接按下 No，也能結束購買。

但比較麻煩的是我寫的形式，如果直接點右上角的叉叉關掉的話，迴圈還會繼續跑，程式就會當掉，我對 form 的運用還不夠深，肯定有辦法及時判斷頁面的開啟 or 關閉狀況，但我不會寫……

```

private void 升級ToolStripMenuItem_Click(object sender, EventArgs e)
{
    upgrade.buy = false;
    upgrade.finished = false;
    upgrade.ShowDialog();
    while (!upgrade.finished)
    {
        while (upgrade.buy == true)
        {
            switch (upgrade.selected)
            {
                case "DoubleTap":
                    if (upgrade.buy && !upgrade.ownedDoubleTap)
                    {
                        if (cookie.cookie >= 100 && cookie.cursorNum >= 1)
                        {
                            cookie.cookie -= 100;
                            cookie.clickCPS += 1;
                            Update();
                            upgrade.ownedDoubleTap = true;
                        }
                        else MessageBox.Show("Requirement not confirmed");
                    }
                    else MessageBox.Show("You already have it.");
                    upgrade.finished = true;
                    upgrade.buy = false;
                    break;
            }
        }
    }
}

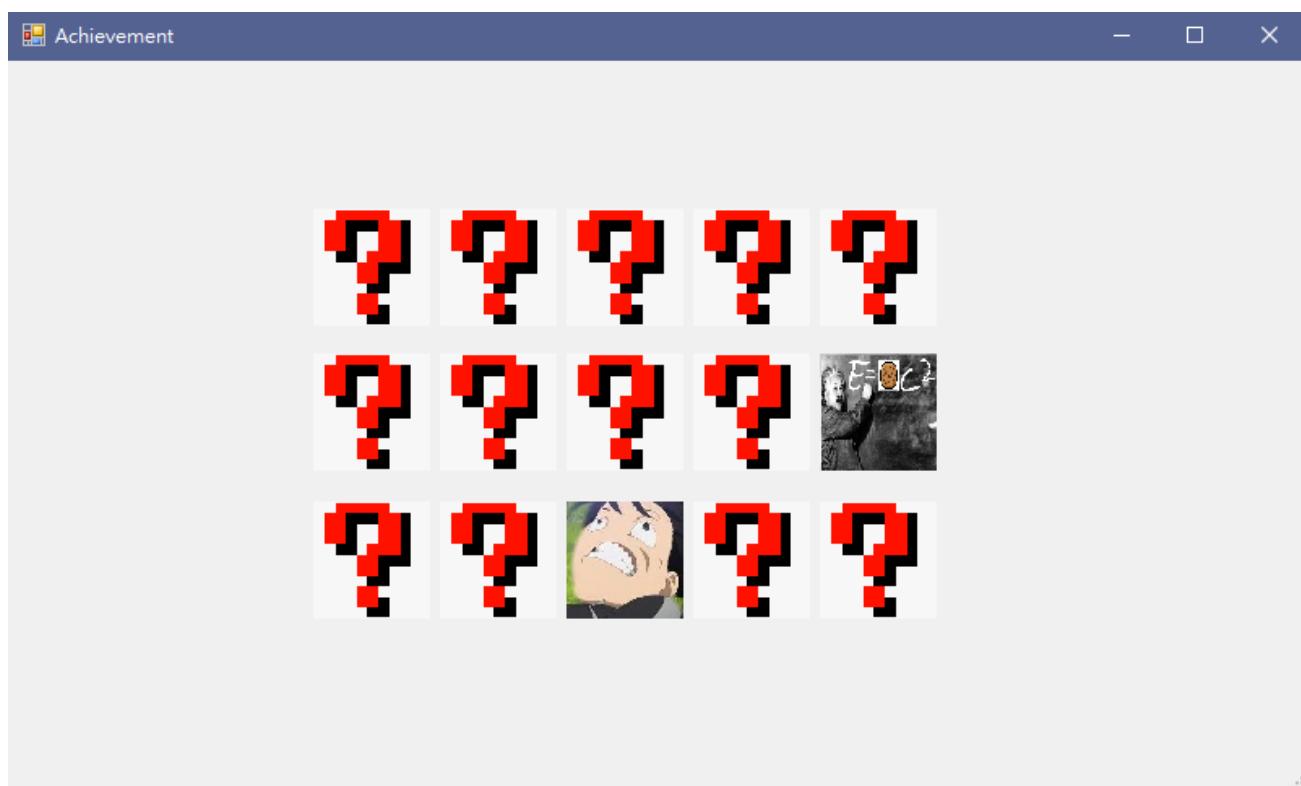
```



五、成就介面：

原作中的成就不只是蒐集用的而已，還會給予餅乾產量百分比加成、影響背景的牛奶口味、並且和許多升級項目有關。

而我設計的成就介面主要為本遊戲的通關目標。



所有未完成的成就，都會顯示問號。達成的時候，在主畫面中會有提示文字以及音效。成就的獲得與否交給 Update 來判斷，也就是每秒更新，因為我不知道 C# 怎麼寫出 once...then... 的這種寫法，個人技術不足(我需要 Python)。

所有的圖片都搭配了 Mouse Hover 的判斷，當滑鼠移到上面就可以看到成就敘述了！

六、心得與反思

本次作業中也遇到了許多多問題，邊玩又跑出一堆 BUG，要一直做調整。不過後來有交給三四個同學做測試，(希望)應該不會有 BUG。

而且每個 Form 都有他的程式碼，我又有另外開幾個類別，還不習慣這種方法的我，切來切去的結果，就是一直寫錯東西，又或是不小心把一堆東西寫在同一個地方，讓程式變得很冗長。