



今天吃甚麼~ & 十八啦~

windowform

作業二

張采翔 | 計算機程式設計 | 8/5

目錄

選擇題目之動機-----	2
十八啦遊戲規則-----	2
構想解說-----	2
結構圖-----	4
whattoeat 程式列表-----	5~6
十八啦程式列表-----	7~8
程式測試規劃-----	9
程式執行結果-----	10~13
參考文獻-----	13

選擇題目之動機

因為最近疫情每天待在家不知道要吃甚麼，而且我家住得很偏僻，每周要去買食材都要開車，所以我都先想要做甚麼菜，為了解決每次煩惱要吃甚麼，所以想寫這個程式來用，也可以因應我個人的需求來設定。最後是因為當我做完十八啦遊戲程式的 window form 以後覺得東西有點少，就來做這個。

十八啦遊戲規則

- 參與者有莊家一人及閒家數人。
- 每局開玩前，閒家先各自下注於自己面前。接著由莊家最後擲骰子，再依序由各閒家逐一擲骰子，以所得點數大小與莊家分勝負。
- 計點方式是先扣除 2 顆點數相同的骰子，其餘 2 顆點數合計即為其所得點數。若有兩組骰子點數相同，則以點數較大者之合計為其所得點數。
 - 因此，所得點數最大為 12 點，稱為「十八」；最小為 3 點，稱為「BG」。
 - 若四顆骰子正面均相同，稱為「一色」（或稱豹子），不論點數多寡，均以勝計。
 - 若所擲骰子正面均不相同，稱為「無面」或「無點」。

構想解說

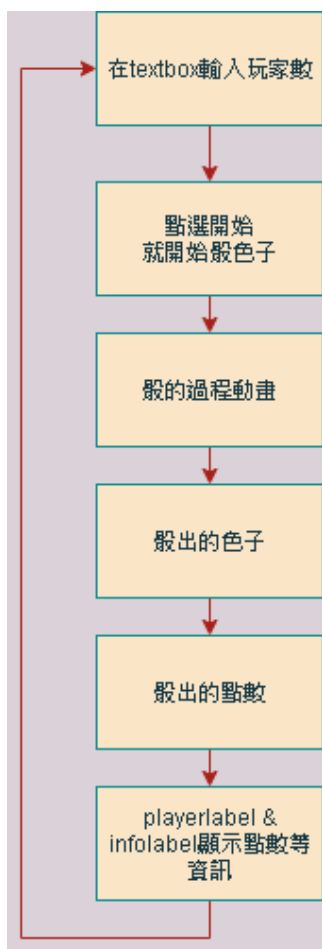
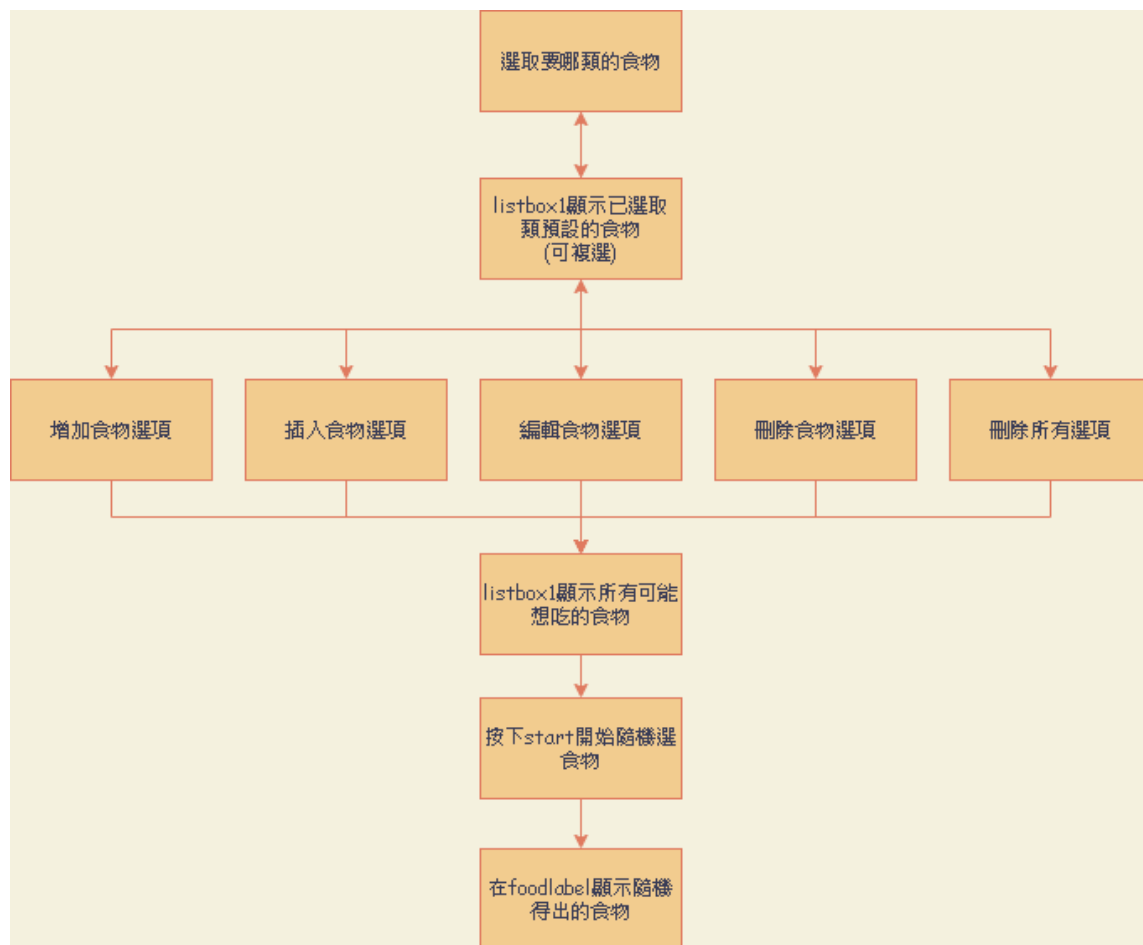
一、WhattoEat 程式

使用者先選要哪類型的食物，在食物清單(listbox)中會自動生成已選取類型的預設食物，再來我們可以再另行編輯此清單，增加、插入、刪除、編輯、刪除全部，等到將自己可能想吃的所有食物都放進清單後，按下 start 就會隨機選取清單內的食物，最後顯示給使用者。

二、十八啦程式

使用者先輸入玩家數目(包含莊家)，輸入完按下開始，色子便會開始骰，每次玩家在骰的時候，都會有一段骰骰子動畫，骰完以後會停留在骰到的色子一下，再來就會顯示玩家骰出的點數，而後再又下一位玩家骰骰子，直到所有玩家都骰完了，便會顯示哪一位玩家獲勝。

WHATTOEAT 結構圖



十八啦結構圖

WHAT TO EAT 程式列表

```
private void Addbutton_Click(object sender, EventArgs e)
{
    listBox1.Items.Add(textBox1.Text);
}
1 個參考
private void Insertbutton_Click(object sender, EventArgs e)
{
    int index = listBox1.SelectedIndex;
    listBox1.Items.Insert(index, textBox1.Text);
}
1 個參考
private void Editbutton_Click(object sender, EventArgs e)
{
    int index = listBox1.SelectedIndex;
    listBox1.Items[index] = textBox1.Text;
}
1 個參考
private void Deletebutton_Click(object sender, EventArgs e)
{
    int index = listBox1.SelectedIndex;
    if(index<0)
    {
        MessageBox.Show("沒有任何項目!");
        return;
    }
    listBox1.Items.RemoveAt(index);
}
```

編輯按鈕事件: 增加、插入、編輯、刪除，若清單內無任何項目就不能刪除。

```

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    int i = comboBox1.SelectedIndex;
    switch (i)
    {
        case 0:
            listBox1.Items.AddRange(new object[] { "牛肉麵", "滷肉飯", "水餃", "炒飯", "炒麵" });
            break;
        case 1:
            listBox1.Items.AddRange(new object[] { "肉醬千層麵", "白醬義大利麵", "南瓜醬義大利麵", "青醬義大利麵", "披薩" });
            break;
        case 2:
            listBox1.Items.AddRange(new object[] { "壽司", "壽喜燒", "豚骨拉麵", "味噌拉麵", "飯糰" });
            break;
        case 3:
            listBox1.Items.AddRange(new object[] { "漢堡", "熱狗", "牛排", "肋排", "波士頓龍蝦" });
            break;
        case 4:
            listBox1.Items.AddRange(new object[] { "泡菜湯", "韓式拌飯", "冷面", "辣炒五花肉", "山東燒雞" });
            break;
        case 5:
            listBox1.Items.AddRange(new object[] { "打拋豬飯", "紅咖哩飯", "椒麻雞飯", "綠咖哩牛", "咖哩蝦" });
            break;
        case 6:
            listBox1.Items.AddRange(new object[] { "珍珠奶茶", "芋頭鮮奶", "珍珠奶綠", "四季春", "阿華田" });
            break;
    }
}

```

當選取食物類型選項為第一個(中式、i==0)在清單內便會顯示牛肉麵、滷肉飯等等預設的選項，並以此類推。

```

private void comboBox1_MouseClick(object sender, MouseEventArgs e)
{
    comboBox1.Text = "";
    comboBox1.DroppedDown = true;
}
1 個參考
private void Startbutton_Click(object sender, EventArgs e)
{
    Random rand = new Random();
    int n = rand.Next(0, listBox1.Items.Count);
    string s = Convert.ToString(listBox1.Items[n]);
    foodlabel.Text = s;
}

```

且當滑鼠在 combobox 點選時，其預設提示字便會消失。

按下 start 隨機選取清單內食物，隨機亂數用 listBox.items.count 得出有幾項，最後在 foodlabel 顯示結果。

十八啦程式列表

```
private void Dicebutton_Click(object sender, EventArgs e)
{
    Dicebutton.Visible = false;
    int n = int.Parse(PeopletextBox.Text);
    int[] x = new int[4];
    PictureBox[] pp = { pictureBox1, pictureBox2, pictureBox3, pictureBox4 };
    string player = "各玩家的點數: \n";
    Random rnd = new Random();
    int[] score = new int[n];
    string info = "";
```

按下開始按鈕後，第一宣告一些基本的參數，包括玩家數，picturebox 要用的 array 和四顆骰子骰出的點數，每個玩家的點數存在 score[]。設置一個亂數產生器作為隨機骰出點數而用。另外使開始按鈕消失(dicebutton)。之後便進入 while 自動跑出每一位玩家骰骰子的活動。

```
Console.ReadLine();
int i = 0;
while (i < n)
{
    for (int k = 0; k < 5; k++) // round
    {
        for (int j = 1; j < 7; j++)
        {
            pp[0].Image = Image.FromFile((j) % 6 + 1 + ".png");
            pp[1].Image = Image.FromFile((j) % 6 + 1 + ".png");
            pp[2].Image = Image.FromFile((j) % 6 + 1 + ".png");
            pp[3].Image = Image.FromFile((j) % 6 + 1 + ".png");
            Refresh();
            Thread.Sleep(50);
        }
    }

    for (int j = 0; j < 4; j++)
    {
        x[j] = rnd.Next(1, 7);
        pp[j].Image = Image.FromFile(x[j] + ".png");
    }
    Refresh();
}
```

第二用 for 迴圈製造動畫的感覺，藉由用 Refresh()和 thread.sleep 讓其圖片每次都顯示才會有動畫感，動畫跑完後，就要骰出玩家的色子，由 x 值得出所得每顆骰子點數，並由此顯示其骰子圖案，且一樣要 Refresh()才會顯示出圖片。

```

    Array.Sort(x);
    int ba = Convert.ToInt32(x[0] == x[1]);
    int bb = Convert.ToInt32(x[1] == x[2]);
    int bc = Convert.ToInt32(x[2] == x[3]);
    int B = ba + bb * 2 + bc * 4;
    switch (B)
    {
        case 0:
            info = "數字皆不同 ";
            score[i] = 0;
            break;
        case 3:
            score[i] = x[0] + x[3];
            break;
        case 6:
            score[i] = x[0] + x[3];
            break;
        case 1:
            score[i] = x[2] + x[3];
            break;
        case 2:
            score[i] = x[0] + x[3];
            break;
        case 4:
            score[i] = x[0] + x[1];
            break;
        case 5:
            if (x[1] > x[2]) score[i] = 2 * x[1];
            else score[i] = 2 * x[2];
            break;
        case 7:
            info = "一色 ";
            score[i] = 12;
    }

```

最後判斷目前這位玩家骰出點數為幾點，陣列在排序後， $x[0] \leq x[1] \leq x[2] \leq x[3]$ ，只需要考慮 8 種情況，再來用 B 這個運算得出 0~7，這八個值便為八種情況，最後算出玩家所得點數。

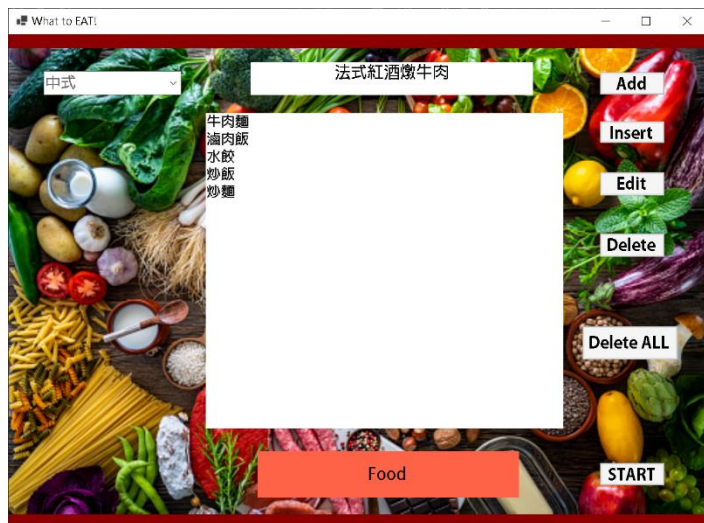
WHATTOEAT 程式測試規劃

- 1.選取食物類別後，食物清單內是否有顯示此類預設的食物
- 2.是否可以在清單內增加食物
- 3.是否可以在清單內編輯食物
4. 是否可以在清單內插入食物
- 5.是否可以在清單內刪除食物
6. 是否可以刪除所有在清單內的食物
- 7.按下 start 是否有隨機從清單內選出食物並顯示在 foodlabel

十八啦程式測試規劃

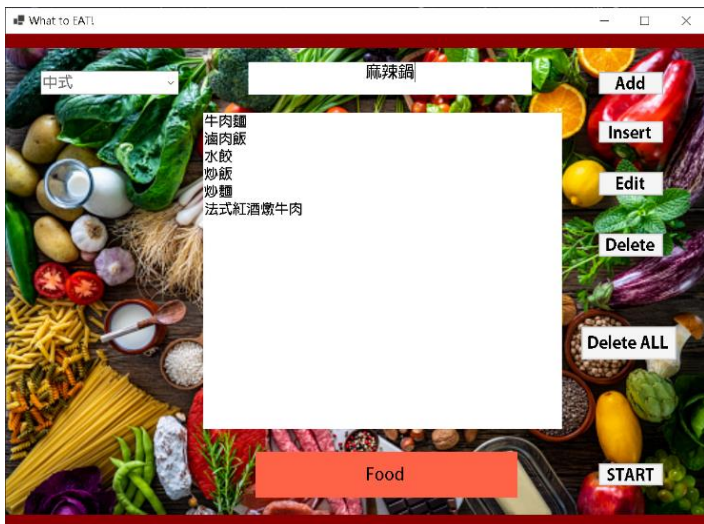
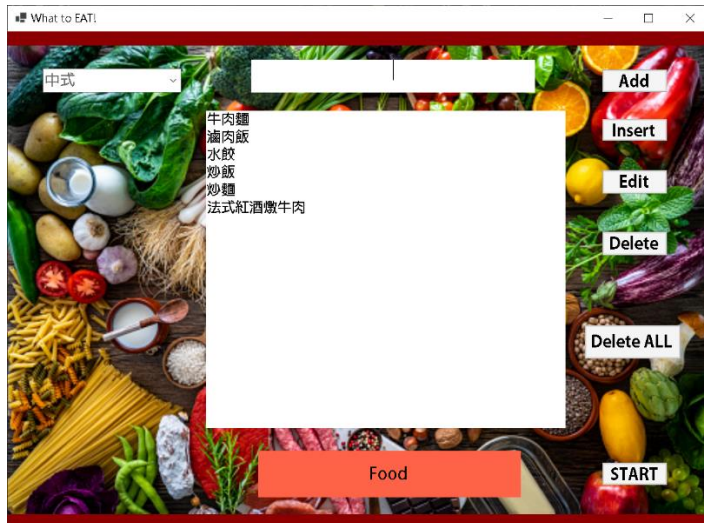
- 1.骰骰子動畫是否完整呈現
- 2.每位玩家骰出的值是否符合骰子圖
- 3.每次骰玩是否有停一下
- 4.所得點數是否完整呈現在畫面中
- 5.此局遊戲最終獲勝者為哪位玩家。

WHATTOEAT 程式執行結果



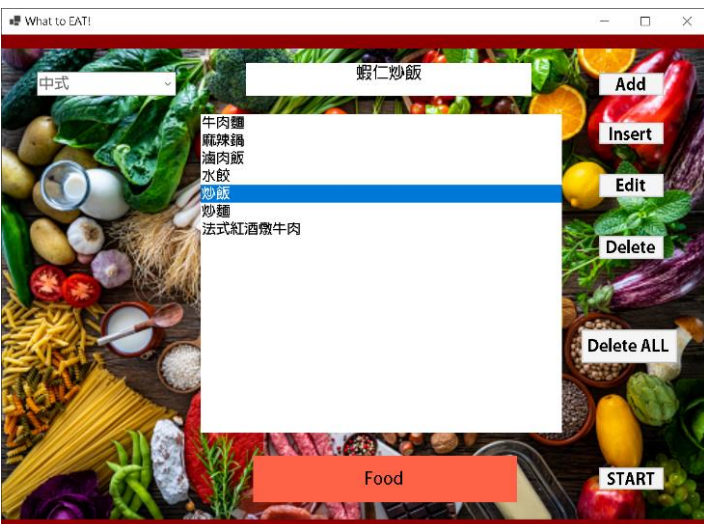
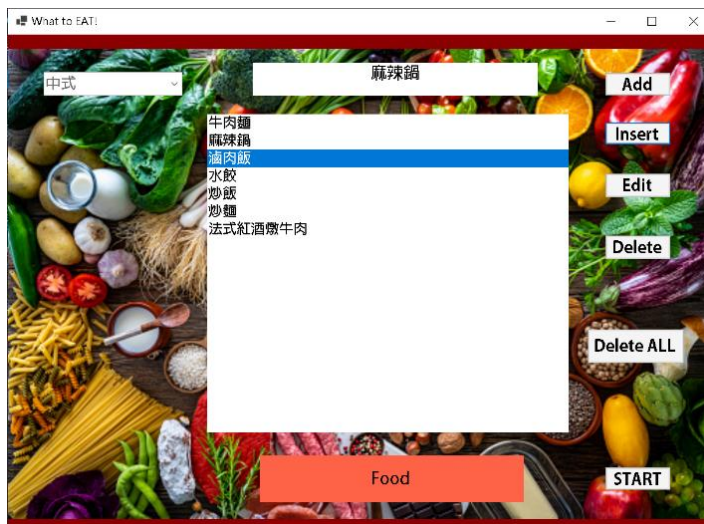
增加
按鈕
功能
測試

完成!



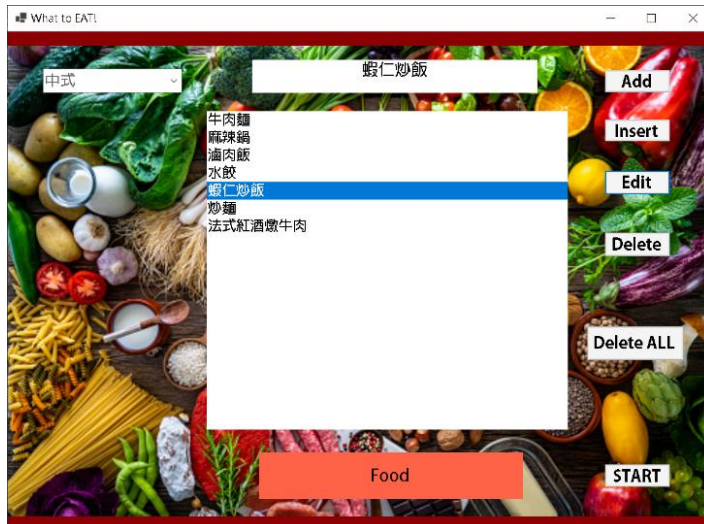
插入
按鈕
功能
測試

完成!



編輯
按鈕
功能
測試

完成!

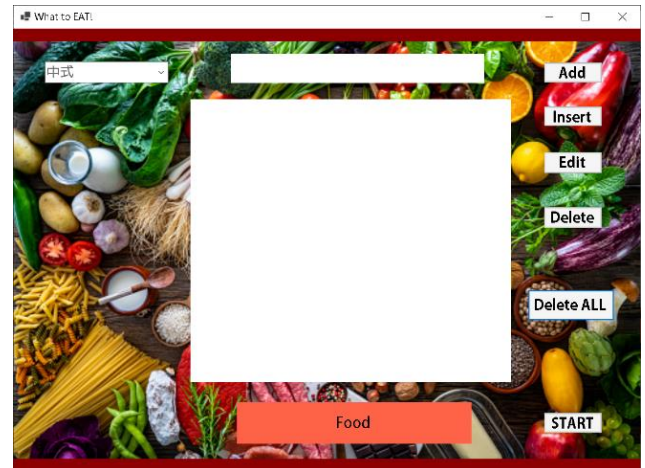


左圖為刪除單一項按鈕功能測試



右圖為刪除所有按鈕測試

皆完成!



Start 隨機選取按鈕測試

完成!

所有功能接測試完畢，且達到預期效果。

十八啦程式執行結果

Input: 5

Output: 測試場景 1 下，執行程式後的輸出如圖 1 所示。

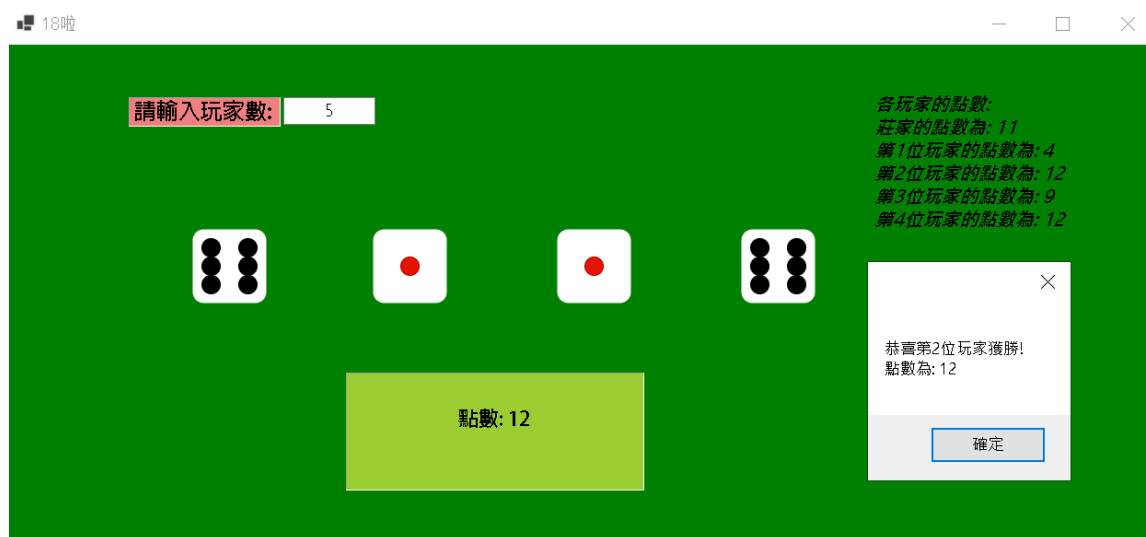


Figure 1

Input: 3

Output: 測試場景 2 下，執行程式後的輸出如圖 2 所示。

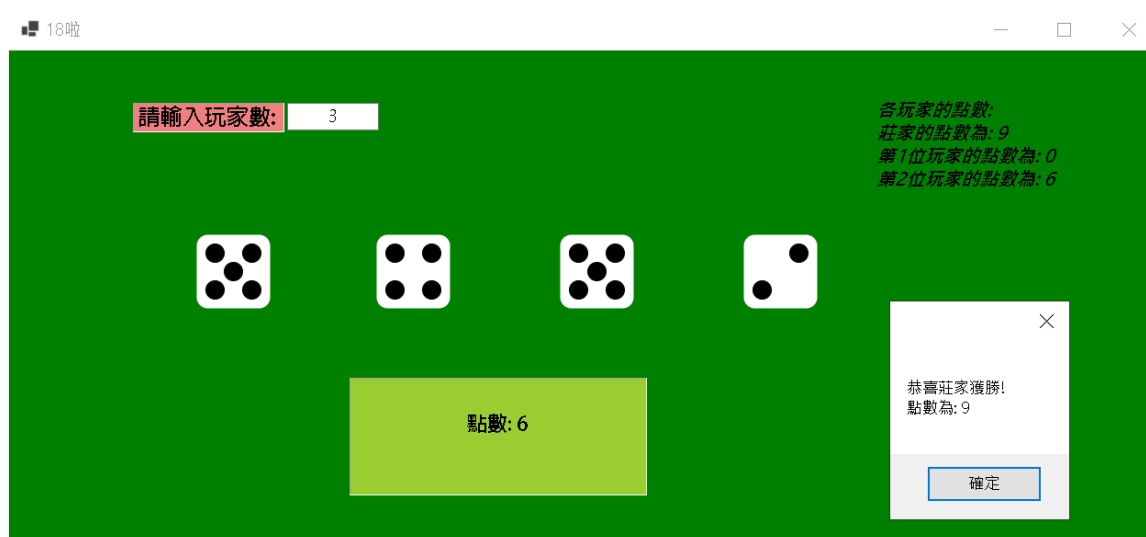


Figure 2

Input: 10

output: 測試場景 3 下，執行程式後的輸出如圖 3 所示。

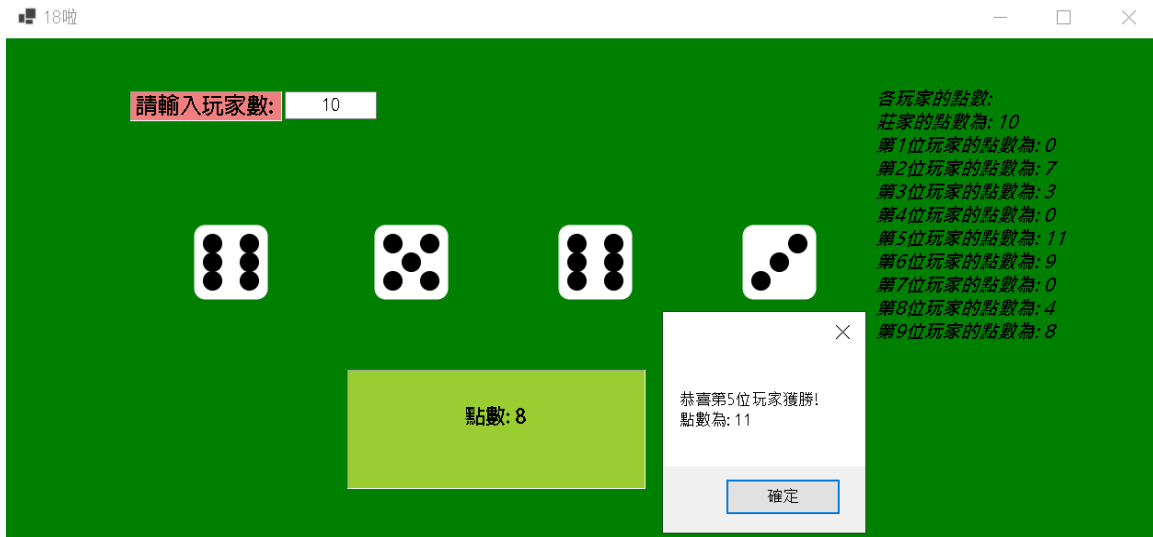


Figure 3

程式結果達成預期目標，且合理。

參考資料

[十八仔 - 維基百科，自由的百科全書 \(wikipedia.org\)](https://zh.wikipedia.org/)

https://www.youtube.com/watch?v=Mf_VwVB_bV8&t=67s

<https://docs.microsoft.com/zh-tw/dotnet/csharp/>