

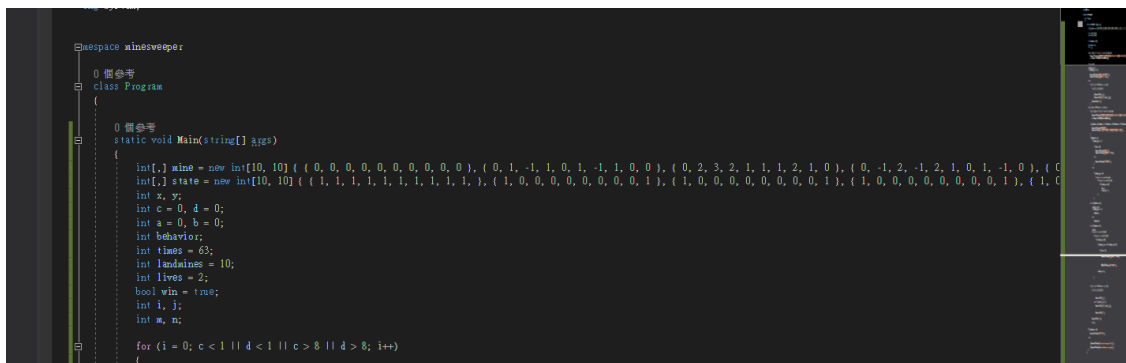
踩地雷—兩個初學者的嘗試

1 選擇題目之動機

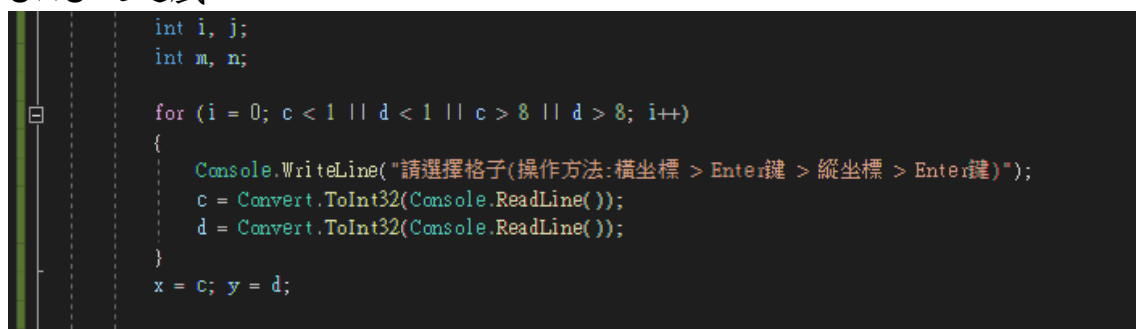
十分喜歡玩踩地雷的遊戲，於是想要建構出類似的程式，作為期末專題，此為其先期嘗試。雖然身為程式新手，做的或許不盡如意。

****規則補充:**踩地雷的遊戲規則相當簡單，在一張地圖上會有許多小方塊需要點開，點開小方塊後會顯示一個數字，這個數字就代表著以它為中心的九宮格內藏著幾顆地雷，只要點到地雷就輸了。踩地雷的獲勝條件是點開所有安全方塊，遊戲中也有「標記」功能可以標記地雷來避免自己誤觸。

2 構想解說



此程式是由兩個陣列所構成的，mine 用來表示是否有地雷以及附近地雷數，state 代表格子的狀態。我們用 10×10 之陣列來做 8×8 之遊戲。



此迴圈用以確認輸入的格式吻合 8×8 之範圍內。

```

state[x, y] = 1;
if (mine[x, y] == -1)
{
    lives--;
    Console.WriteLine("BOOM!!!\nSO LUCKY!!!");
    Console.WriteLine("剩餘血量:" + lives);
}
else if (mine[x, y] == 0)
{
    for (m = x - 1; m < x + 2; m++)
    {
        for (n = y - 1; n < y + 2; n++)
        {
            if (state[m, n] == 0)
            {
                times--;
                state[m, n] = 1;
            }
        }
    }
    for (i = 1; i < 9 && win == true; i++)
    {
        for (j = 1; j < 9; j++)
        {
            if (state[i, j] == 0)
            {
                Console.Write("_");
            }
            else
            {
                Console.Write("{0}", mine[i, j]);
            }
        }
        Console.Write("\n");
    }
}
else

```

將輸入之第一格開啓，並按照是否有地雷產生不同結果。

```

    Console.WriteLine("\n");
}
for (; times > 0 && win == true; times--)
{
    for (i = 0; a < 1 || b < 1 || a > 8 || b > 8; i++)
    {
        Console.WriteLine("請選擇格子(操作方法:橫坐標 > Enter鍵 > 縱坐標 > Enter鍵);");
        a = Convert.ToInt32(Console.ReadLine());
        b = Convert.ToInt32(Console.ReadLine());
    }
    x = a; y = b;
    for (behavior = 0; behavior != 1 && behavior != 2 && behavior != 3 && behavior != 4;)
    {
        Console.WriteLine("請選擇動作");
        Console.WriteLine("1:採格子 2:插旗子 3:開九宮格(不含旗子) 4:結束");
        behavior = Convert.ToInt32(Console.ReadLine());
    }
}

```

之後進入主迴圈，選取一個有效座標，並按照輸入執行四種行為。

```

if (behavior == 1)
{
    if (mine[x, y] == -1)
    {
        lives--;
        if (lives > 0)
        {
            Console.WriteLine("BOOM!!!");
            Console.WriteLine("剩餘血量:" + lives);
        }
        else
        {
            win = false;
            Console.WriteLine("YOU LOSE!!!");
        }
    }
    else
    {
        state[x, y] = 1;
        if (mine[x, y] == 0)
        {
            for (m = x - 1; m < x + 2; m++)
            {
                for (n = y - 1; n < y + 2; n++)
                {
                    if (state[m, n] == 0)
                    {
                        times--;
                        state[m, n] = 1;
                    }
                }
            }
        }
    }
}

```

第一種行為是將一個格子轉為開啓，並按照是否有地雷產生不同結果。

```

}
else if (behavior == 2)
{
    state[x, y] = 2;
    if (mine[x, y] == -1)
    {
        landmines--;
    }
    else
    {
        landmines++;
    }
}

```

第二種是將一個格子標記為插旗(用來標示地雷位子)。

```

else if (behavior == 3)
{
    times++;
    for (m = x - 1; m < x + 2; m++)
    {
        for (n = y - 1; n < y + 2; n++)
        {
            if (state[m, n] == 0)
            {
                times--;
                if (mine[m, n] == -1 && state[m, n] == 0)
                {
                    lives--;
                    if (lives > 0)
                    {
                        Console.WriteLine("BOOM!!!");
                        Console.WriteLine("剩餘血量:" + lives);
                    }
                    else
                    {
                        win = false;
                        Console.WriteLine("YOU LOSE!!!");
                    }
                }
                state[m, n] = 1;
            }
        }
    }
}

```

第三種為將周圍九宮格之格子轉為開啓狀態(除了插旗的格子)，按照是否有地雷產生不同結果。

```

else if (behavior == 4)
{ times = -4; }
for (i = 1; i < 9 && win == true; i++)
{
    for (j = 1; j < 9; j++)
    {
        if (state[i, j] == 0)
        {
            Console.Write("_");
        }
        else if (state[i, j] == 1)
        {
            Console.Write("{0}", mine[i, j]);
        }
        else if (state[i, j] == 2)
        {
            Console.Write("*");
        }
    }
    Console.Write("\n");
    a = 0;
    b = 0;
}

```

第四種為結束遊戲，我們在程式中有限定執行的次數，若按照正常遊戲方式並不會用完，而選擇此動作可將次數減為負數。

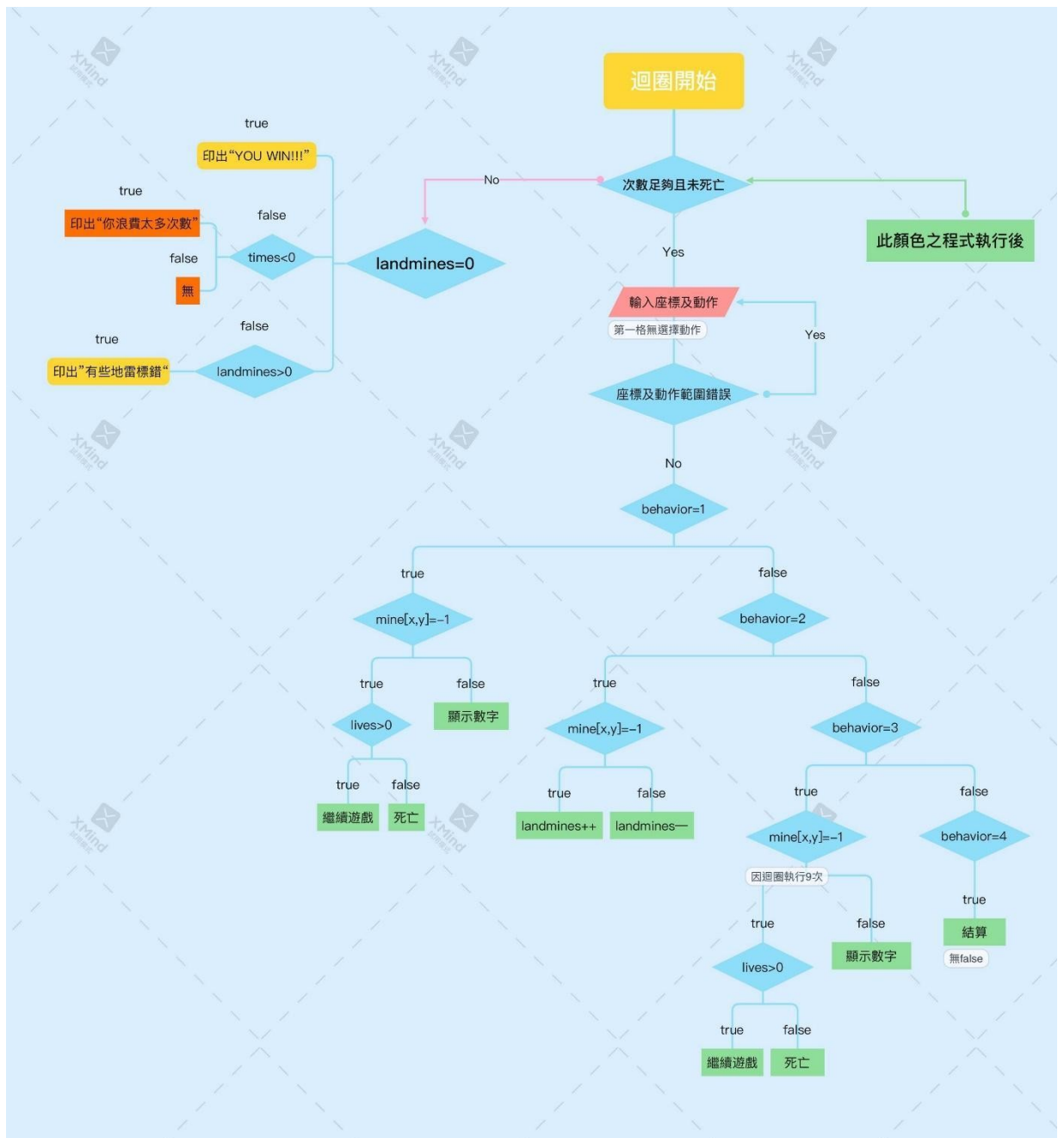
```

}
if (landmines == 0)
{
    Console.WriteLine("YOU WIN!!!");
}
else
{
    if (times < 0)
    { Console.WriteLine("You waste too many times!"); }
    if (landmines > 0)
    { Console.WriteLine("Some landmines are wrong!"); }
}
}

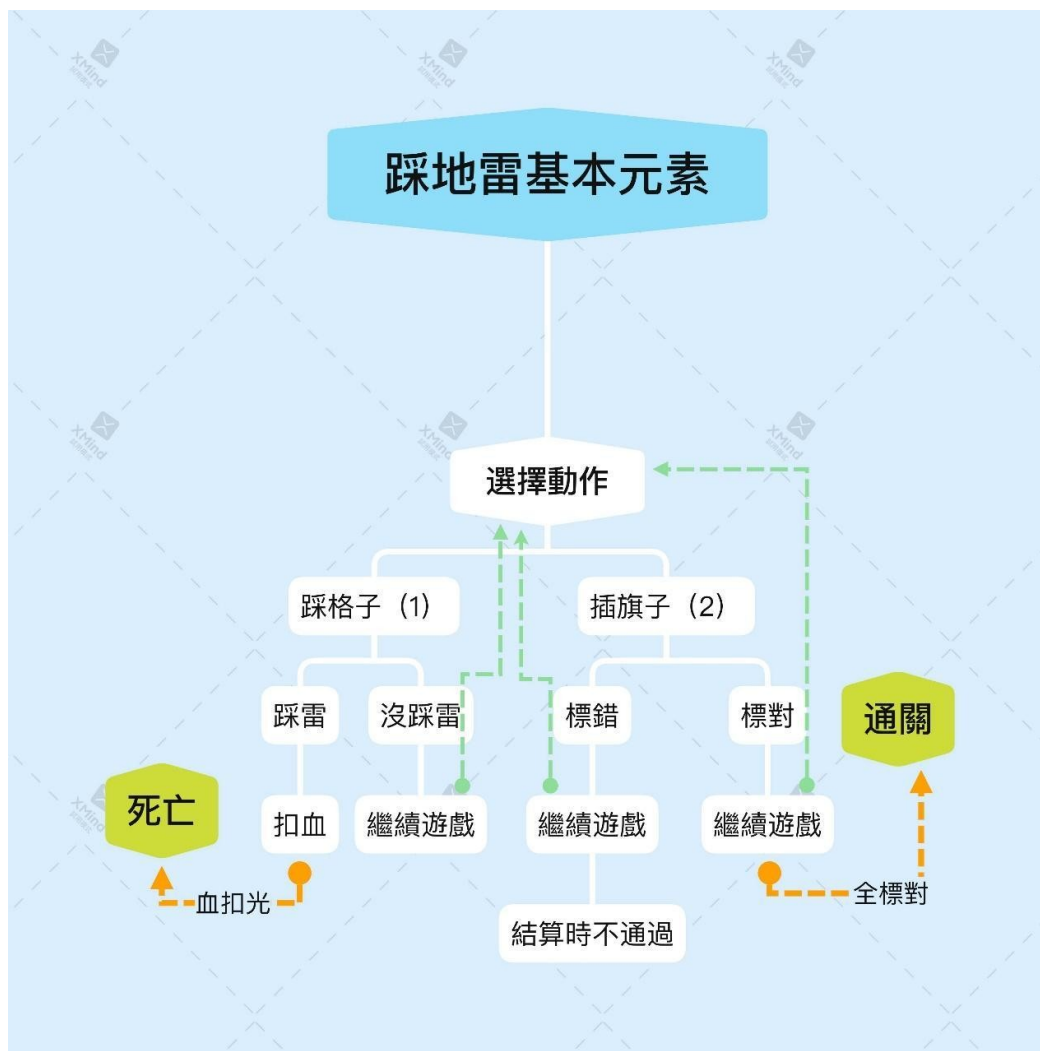
```

最後的這一個部分為遊戲的獲勝及結束的敘述。
雖然真正的踩地雷是將所有非地雷格子開啓而獲勝，我們設計只要將旗子插完且皆正確即可勝利。

3 流程圖



4 結構圖



5 程式測試執行結果

```
C:\WINDOWS\system32\cmd.exe
6
4
請選擇動作
1:踩格子 2:插旗子 3:開九宮格(不含旗子) 4:結束
3
-----
-----
-----
-----
22
211 *1
111211
211*100
1011100
請選擇格子(操作方法:橫坐標 > Enter鍵 > 縱坐標 > Enter鍵)
```

遊戲畫面

```
C:\WINDOWS\system32\cmd.exe
4
請選擇動作
1:採格子 2:插旗子 3:開九宮格(不含旗子) 4:結束
4
1*101*10
23211121
*2*2101*
122*1122
112111*1
1*111211
2211*100
*1011100
YOU WIN!!!
```

通關

```
C:\WINDOWS\system32\cmd.exe
請選擇格子(操作方法:橫坐標 > Enter鍵 > 縱坐標 > Enter鍵)
3
1
BOOM!!!
SO LUCKY!!!
剩餘血量:1
```

第一格即踩雷

6 負責工作

在小組中我負責的工作主要是處理程式碼中迴圈和陣列的部分和繪製流程圖。另一人主要負責測試和抓錯。

7 參考文獻

因為此為先期嘗試，製作的較為簡單，故無參考文獻。預計之後會做較為複雜以及有特殊規則的版本。