



# 雞蛋別想跑

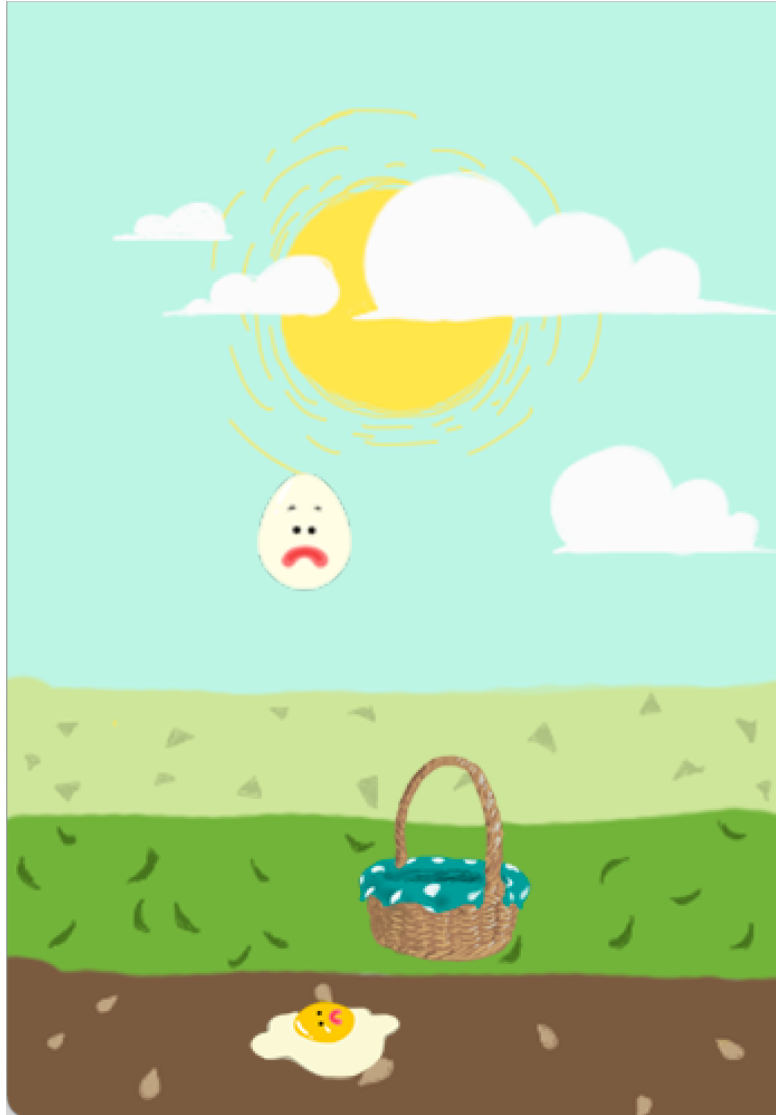


# 拆解

**練習 1-1 籃子動一動**  
籃子跟著游標左右移動，  
但不會超出畫面

**練習 1-2 天雞下蛋**  
雞蛋隨機從畫面上方任一  
位置落下

**練習 2-1 接雞蛋**  
當雞蛋被接到的時候，會  
有另一個再從上方掉落

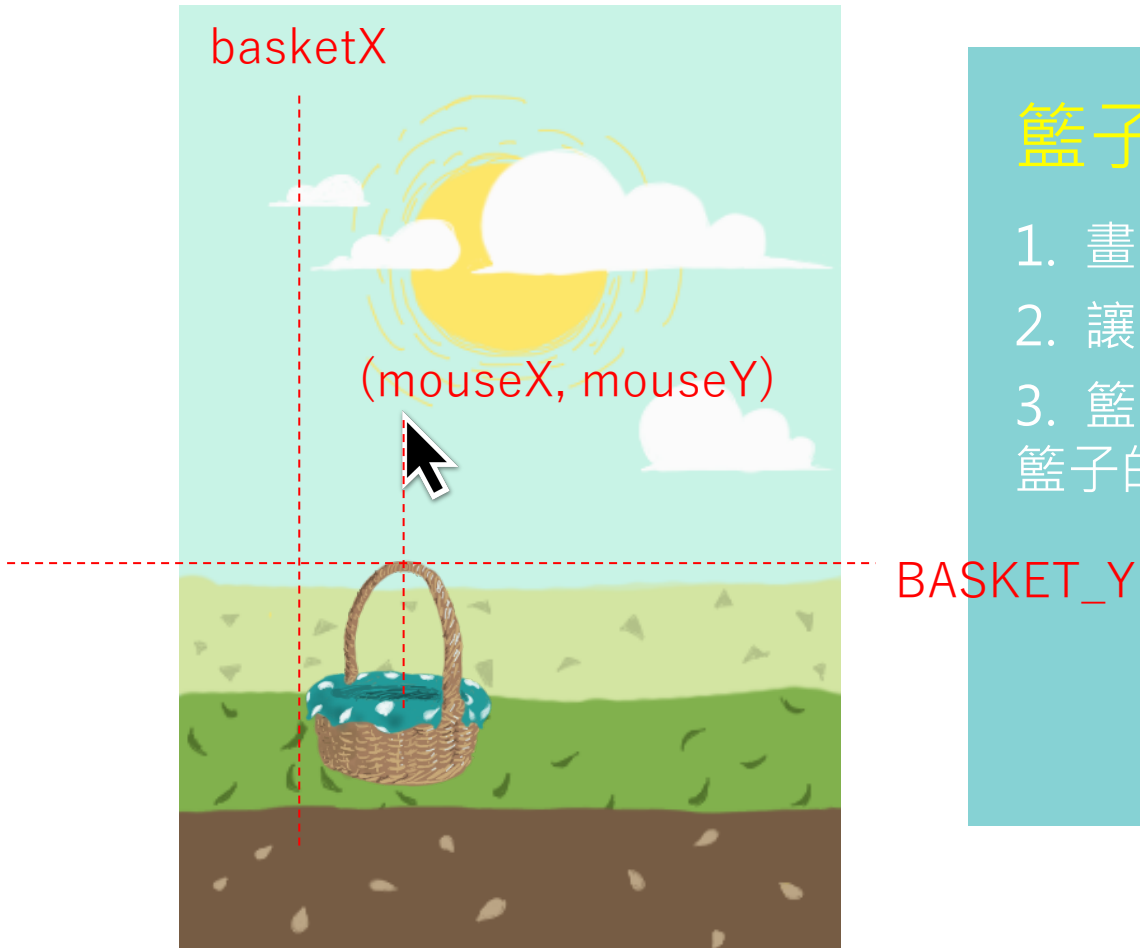


**練習 2-2 破掉的蛋**  
掉在地上的蛋會破掉

**練習 3-1 雞蛋到籃內**  
讓雞蛋掉進籃子「內部」

**練習 3-2 重新開始**  
按下滑鼠，遊戲重新開始

# 練習1-1 籃子動一動



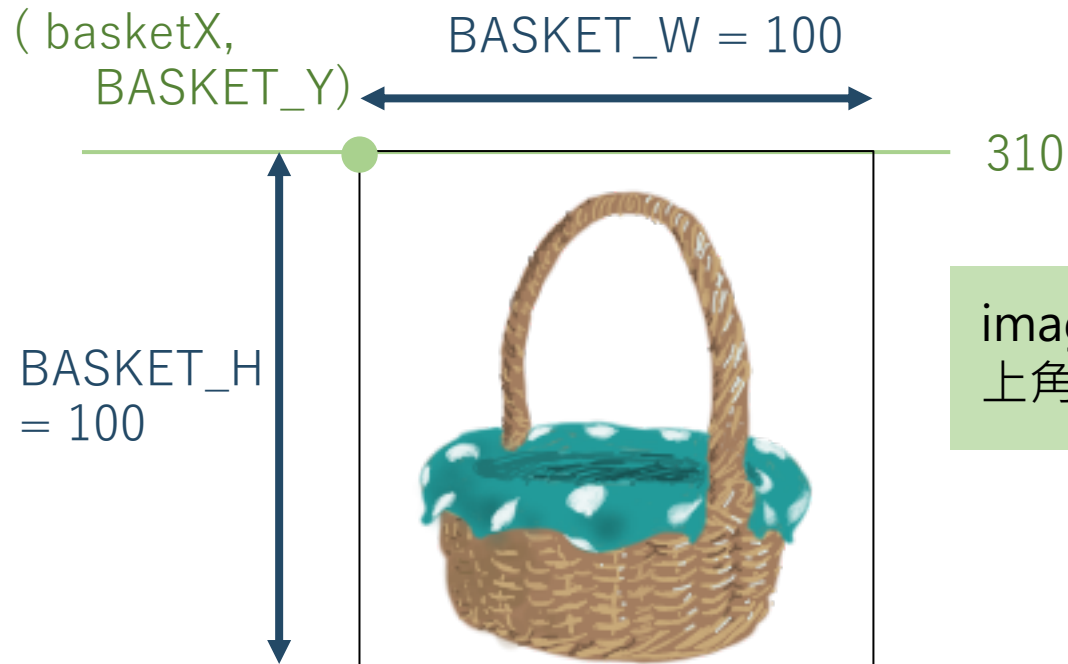
## 籃子動一動的條件

1. 畫布大小：350 x 500。
2. 讓籃子中心跟著游標水平移動，且籃子不會跑出畫面。
3. 籃子的 y座標 固定在 BASKET\_Y。  
籃子的寬度與高度固定為 BASKET\_W 與 BASKET\_H。

# 練習1-1 籃子動一動

籃子要放在哪裡？

```
image ( basket, basketX, BASKET_Y, BASKET_W, BASKET_H );
```



一直都不會改變的數字，可以直接定義成常數。

`image`的位置是以圖片左上角為基準點喔！

# 練習1-1 籃子動一動

籃子移不出畫面

利用 if 來判斷

如果籃子超過左邊界



籃子固定在左邊界

如果籃子超過右邊界



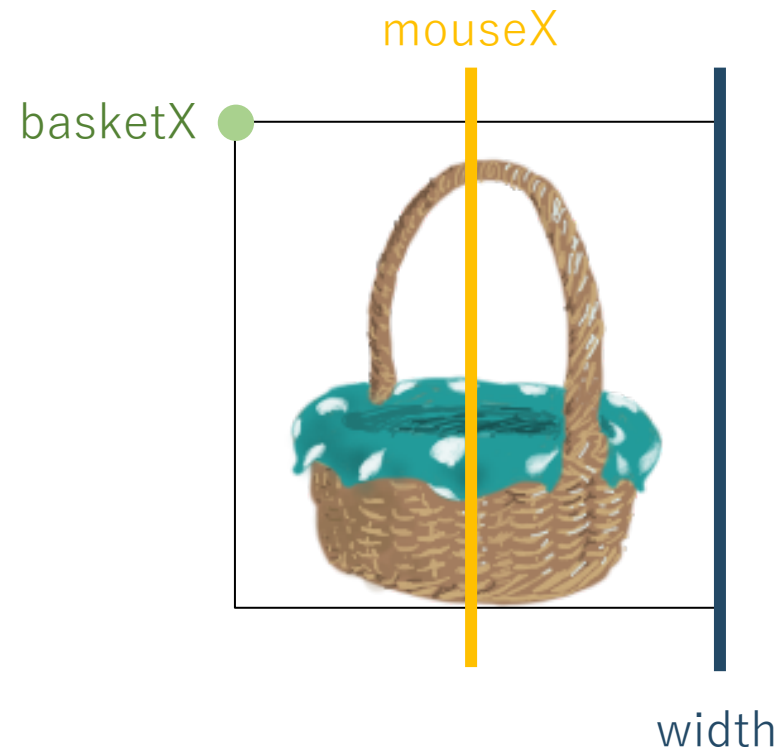
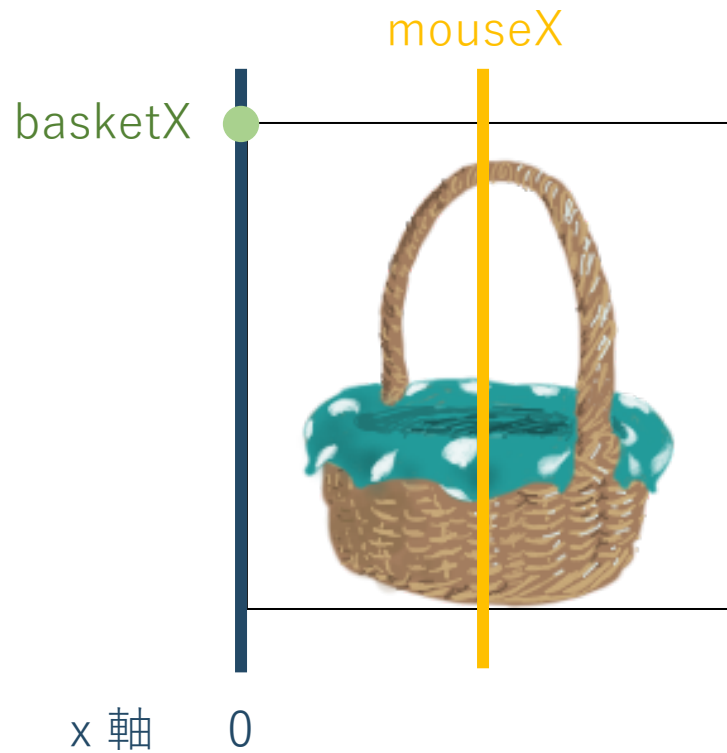
籃子固定在右邊界

注意 !!!!  
要讓籃子完整的呈  
現在畫布上

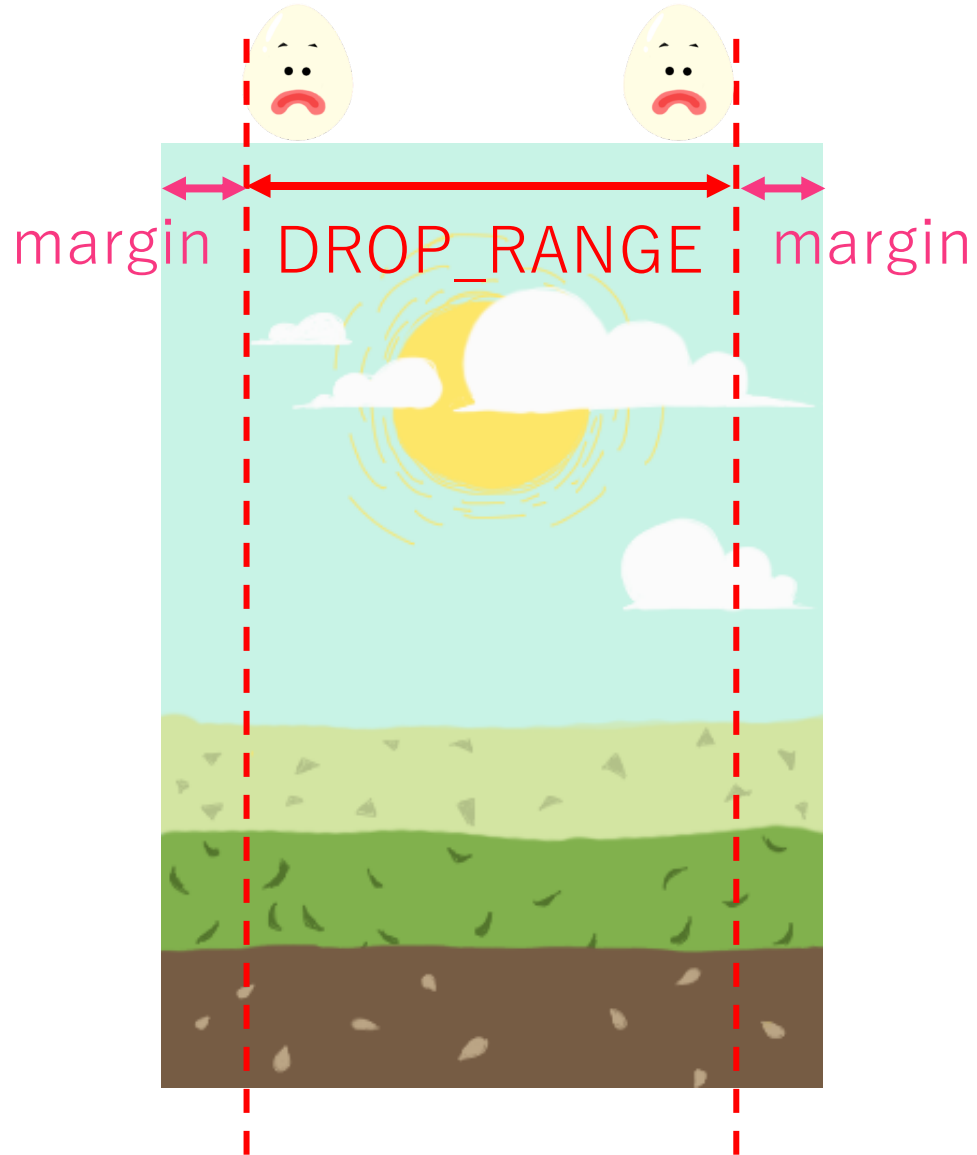
# 練習1-1 籃子動一動

籃子移不出畫面

籃子碰到左右邊界時，完整停在原地不動。



# 練習1-2 天雞下蛋

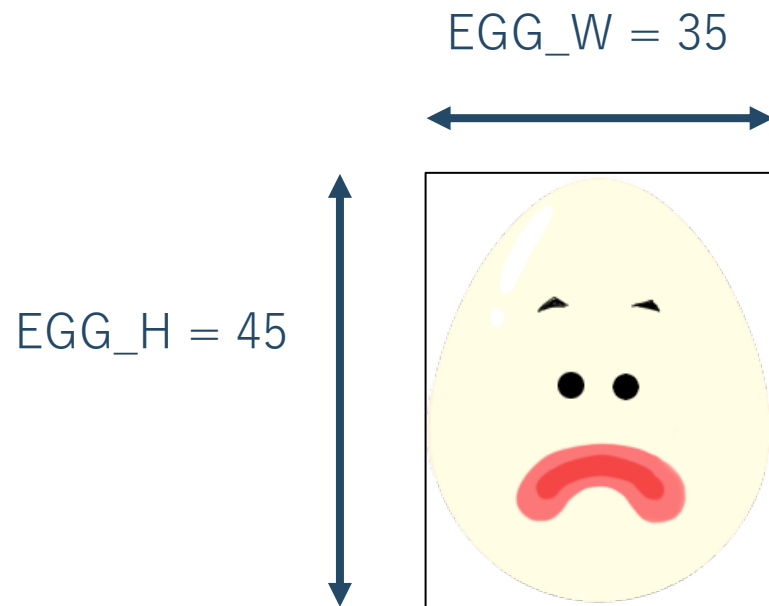


## 天雞下蛋的條件

1. 雞蛋底緣會從畫布頂端出現。
2. 水平下蛋範圍為 `DROP_RANGE`。
3. 蛋的落下速度為 2,3,4 之隨機值。  
(等速度移動; 速度單位: pixel per frame)
4. 蛋的寬度與高度為 `EGG_W`、`EGG_H`。

# 練習1-2 天雞下蛋

雞蛋會在哪裡？





# 練習2-1 接雞蛋



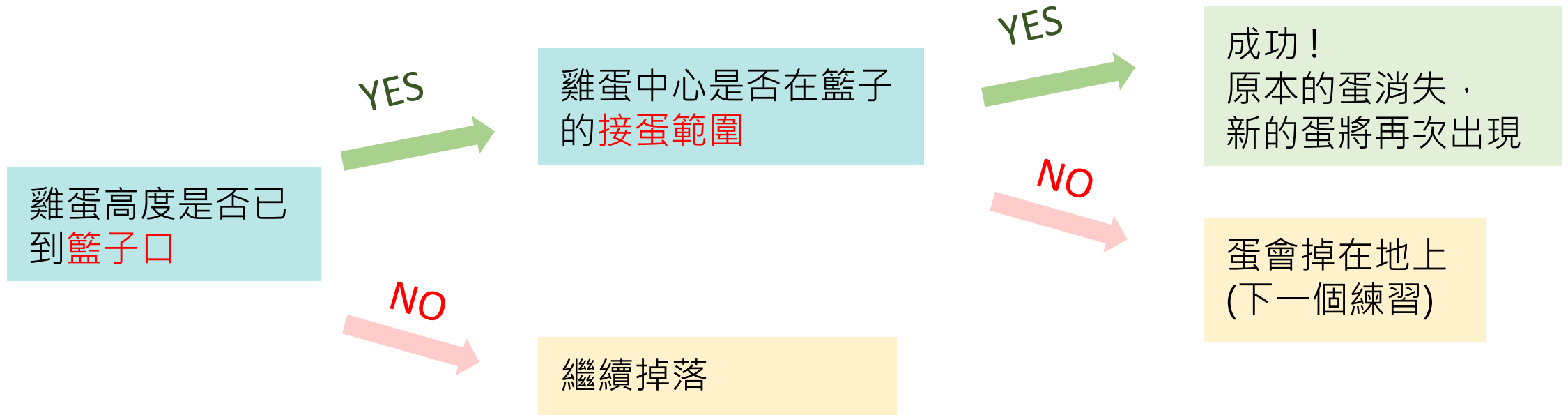
## 接雞蛋的條件

1. 當雞蛋掉進籃子時，表示接到蛋，此時蛋會消失，並從畫布上方出現新蛋。
2. 賦予新的雞蛋新的出發位置及速度。

# 練習2-1 接雞蛋

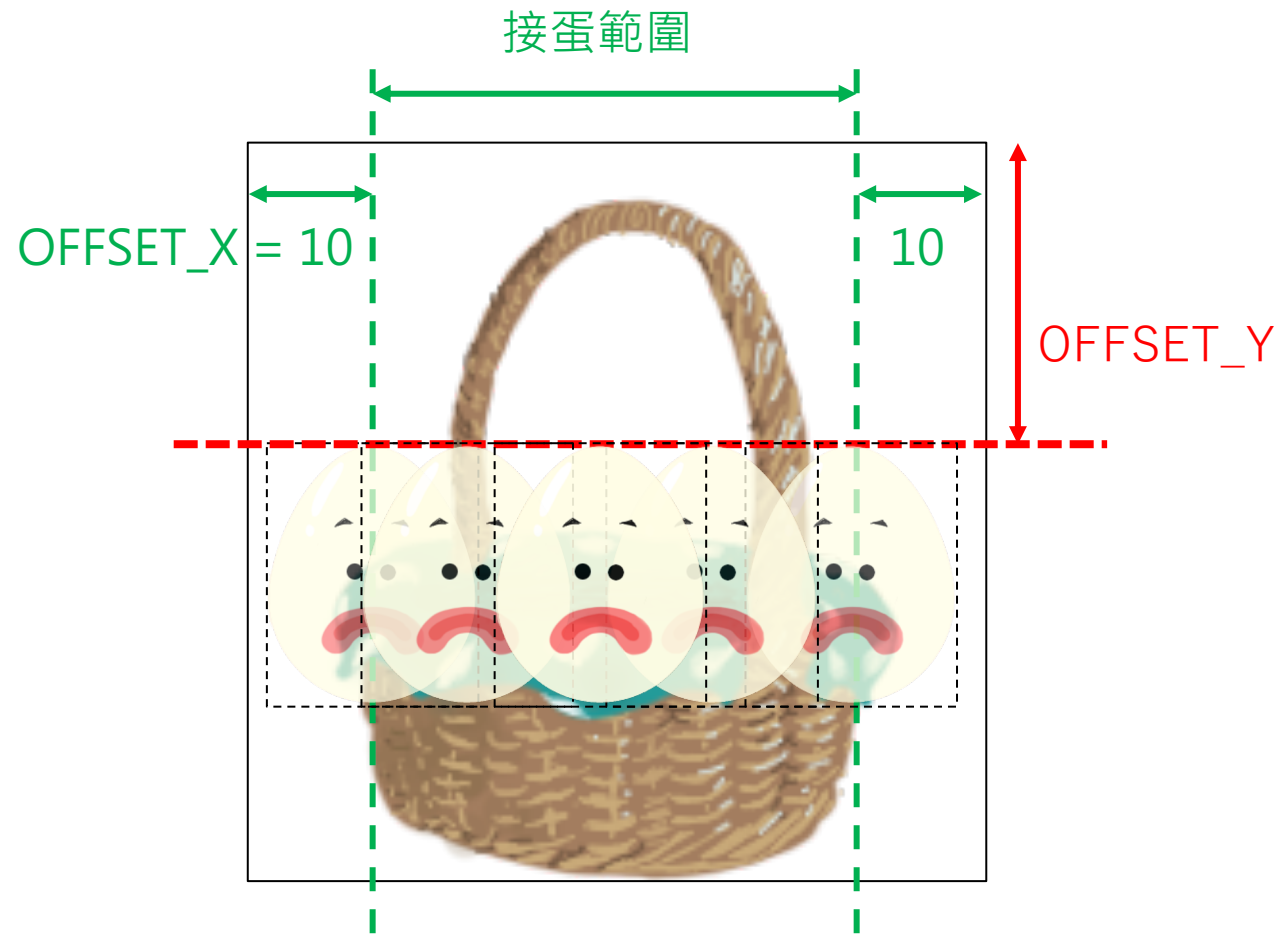
怎樣算成功接到雞蛋？

利用 if 來判斷



# 練習2-1 接雞蛋

接蛋範圍？



# 練習2-1 接雞蛋

```
if ( 雞蛋y 超過基準線 ) {  
    if ( 雞蛋x 在接蛋範圍內 ) {  
        雞蛋重新從上方出現，重新選位置及速度  
    }  
}
```

# 練習2-2 破掉的蛋

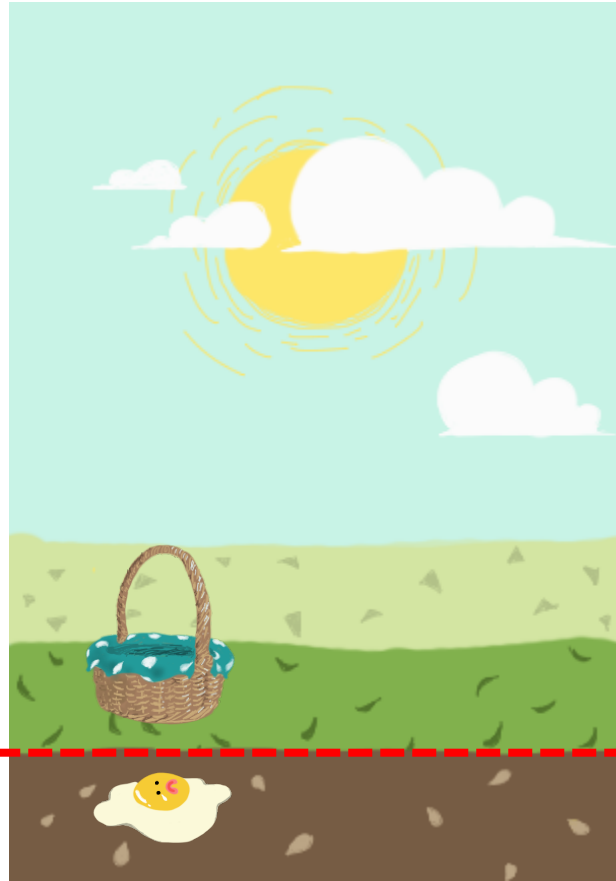


## 破掉的蛋條件

1. 當雞蛋沒被籃子接到，會變成破掉的蛋。
2. 破掉的蛋的  $y$  座標固定在 `GROUND_Y`。
3. 破掉的蛋的  $x$  座標與原掉落的蛋位置相同。
4. 原來的蛋要消失。

# 練習2-2 破掉的蛋

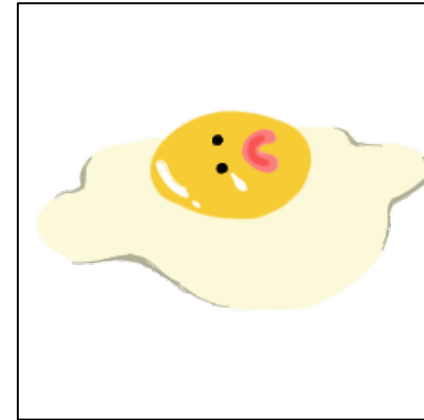
雞蛋墜落的地方



雞蛋墜落位置  
 $\text{GROUND\_Y} = 430$

(原掉落雞蛋的x,  $\text{GROUND\_Y}$ )

$\text{BREAKEGG\_W} = 70$

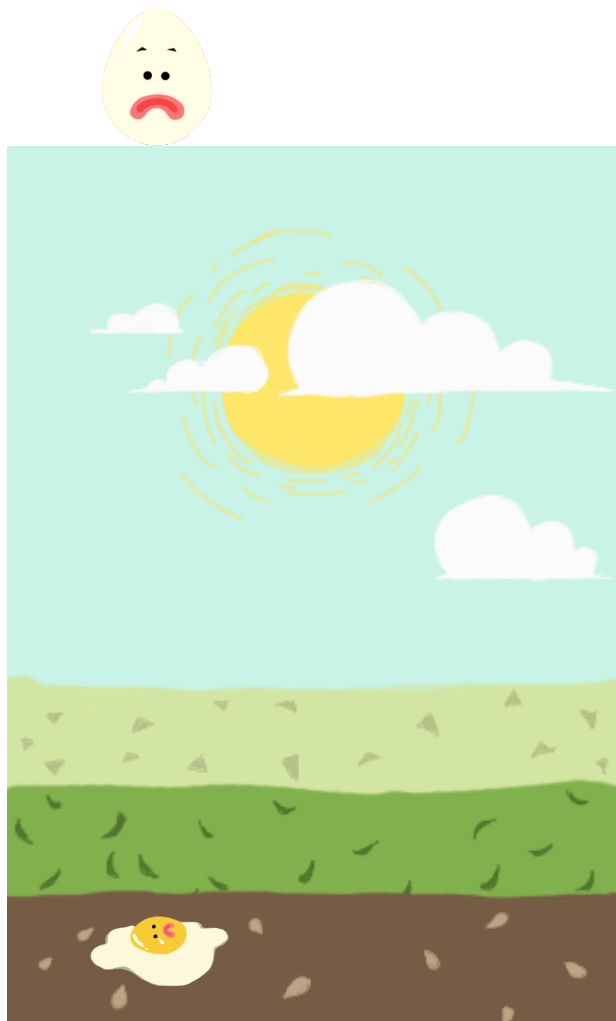


$\text{BREAKEGG\_H} = 70$

# 練習2-2 破掉的蛋

好的雞蛋怎麼消失

把它移出畫面外就可以了



# 練習3-1 雞蛋到籃內



## 雞蛋到籃內的條件

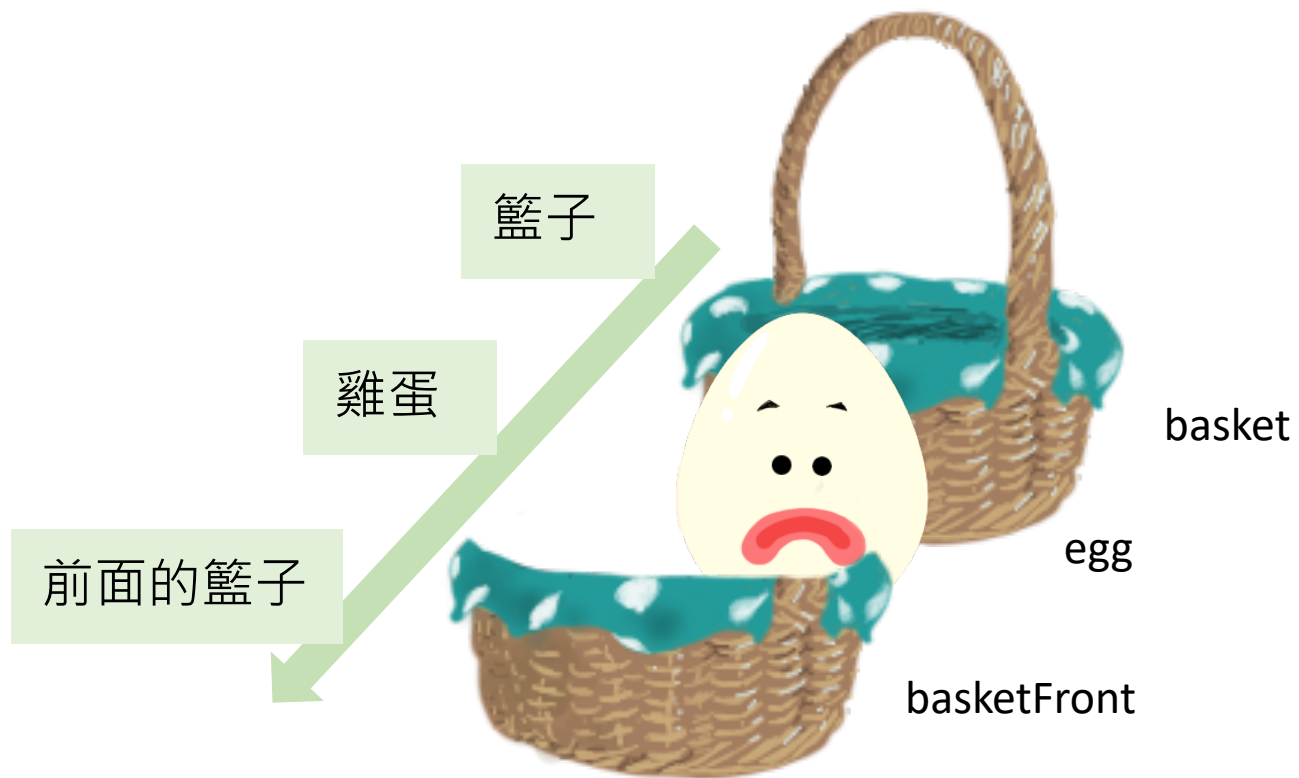
1. 讓雞蛋確實地進到籃子「裡面」。



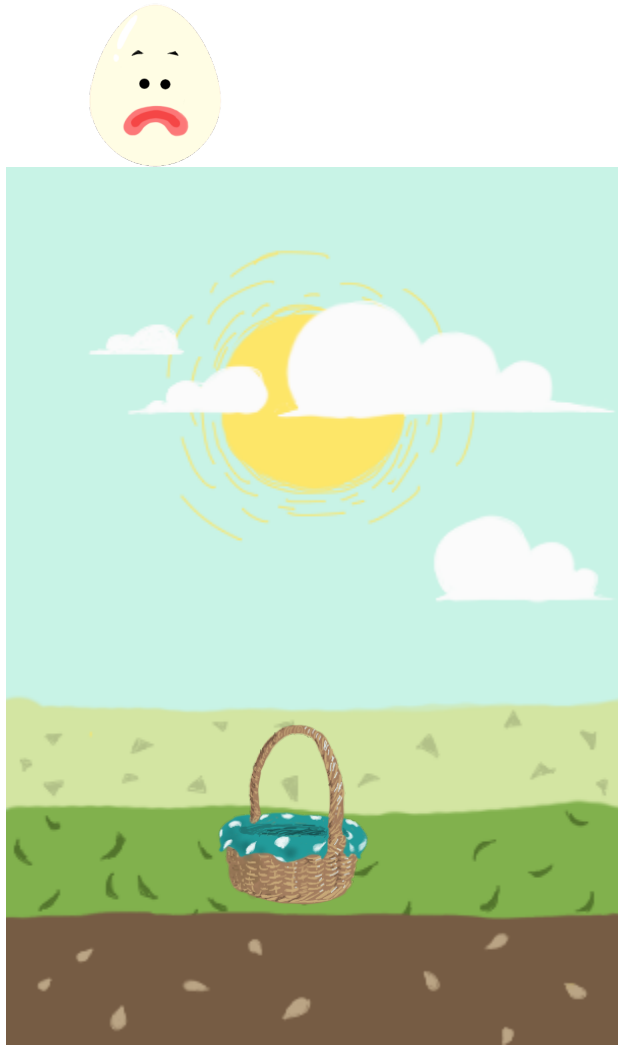
# 練習3-1 雞蛋到籃內

圖層順序

draw() 時考慮各圖層的優先順序



# 練習3-2 重新開始



## 重新開始的條件

1. 蛋破掉之後，按下滑鼠能重新開始遊戲。
2. 重新開始時，蛋要重新選擇落下位置和速度。

# 練習3-2 重新開始

重新開始

draw() 內多一個 if 判斷式

```
if ( !gameOver ) {  
    進行遊戲  
}
```

甚麼情況下會讓  
gameOver 變 true 呢？

Game over 時，Mouse Click 重新開始遊戲