

# Hour of Code @NCCU

政大資科暨數位內容學程 余能豪老師

# 目標

- 完成「搶救食安大作戰」電玩遊戲



# Processing 開發工具

- 由一群藝術家及設計師所開發
- 免費、開放、跨平台
- 適用撰寫影像、動畫、聲音等互動程式
- Processing可直接呈現程式執行後的視覺效果，  
有助於學習程式基礎觀念並讓同學體會程式創作的樂趣。

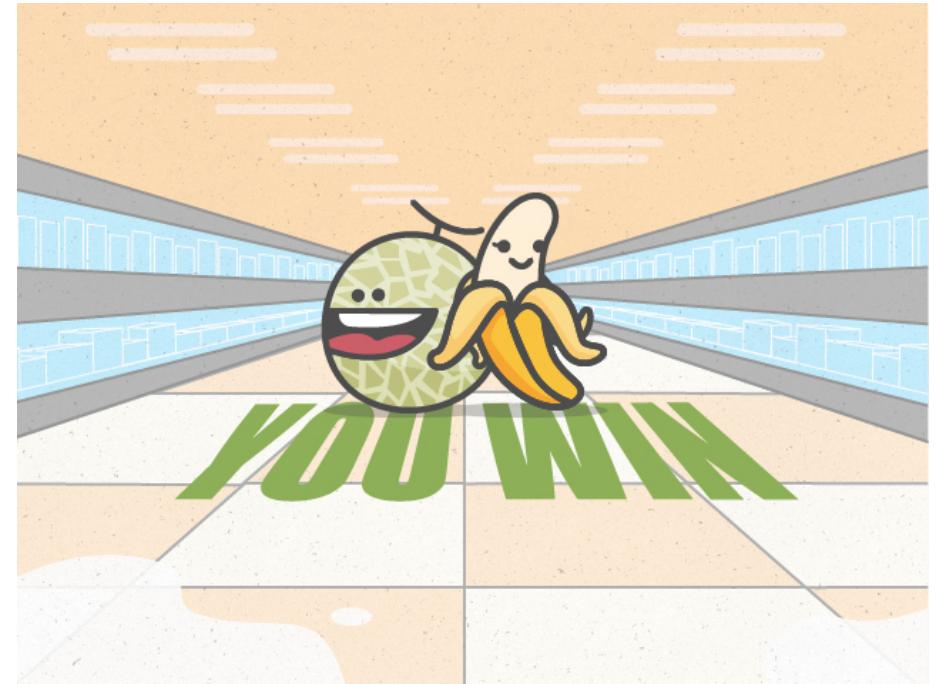
下載連結：<https://processing.org/>

# 今日活動

- 程式基本概念
- 變數、顯示圖片、運算、動畫、亂數
- 事件偵測、判斷式、常數
- 程式結構與流程
- 迴圈
- 陣列
- 自定函式

# 五大關卡

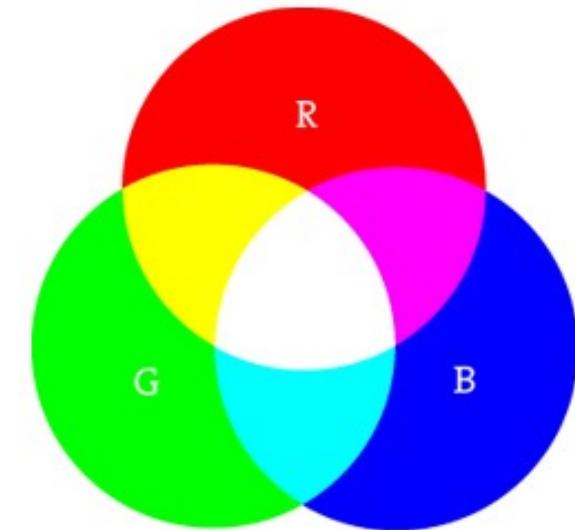
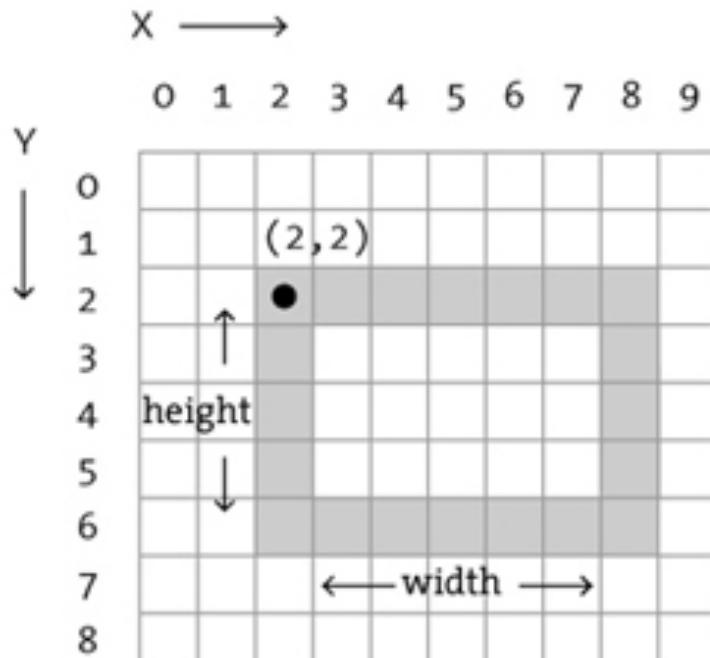
- 鈔票飛上天
  - 牛奶不用錢
  - 水果新樂園
  - 進擊的毒奶
  - 抵制有效
- 
- 累積點數最高的組別將得到獎勵
  - <http://programming101.cs.nccu.edu.tw/hour>



# 何謂電腦程式

- 電腦程式：由一連串的指令所組成，讓電腦按步就班的執行

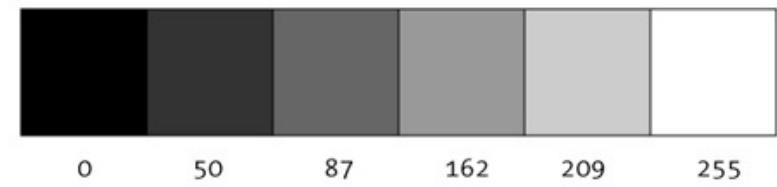
```
size(640,480);
background(50);
fill(255,0,0);
rect(200,200,200,200);
```



`rect(x,y,width,height);`

Example:

`rect(2,2,7,5);`



# 變數 (Variables)

- 宣告變數 Declare the variable

`int myAge;`

Case sensitivity!



- 變數初始化 Initialize the variable

`myAge = 18; // assign value(right) to variable(left)`

Data assignment (assignment operator)



- 使用變數 Use the variable

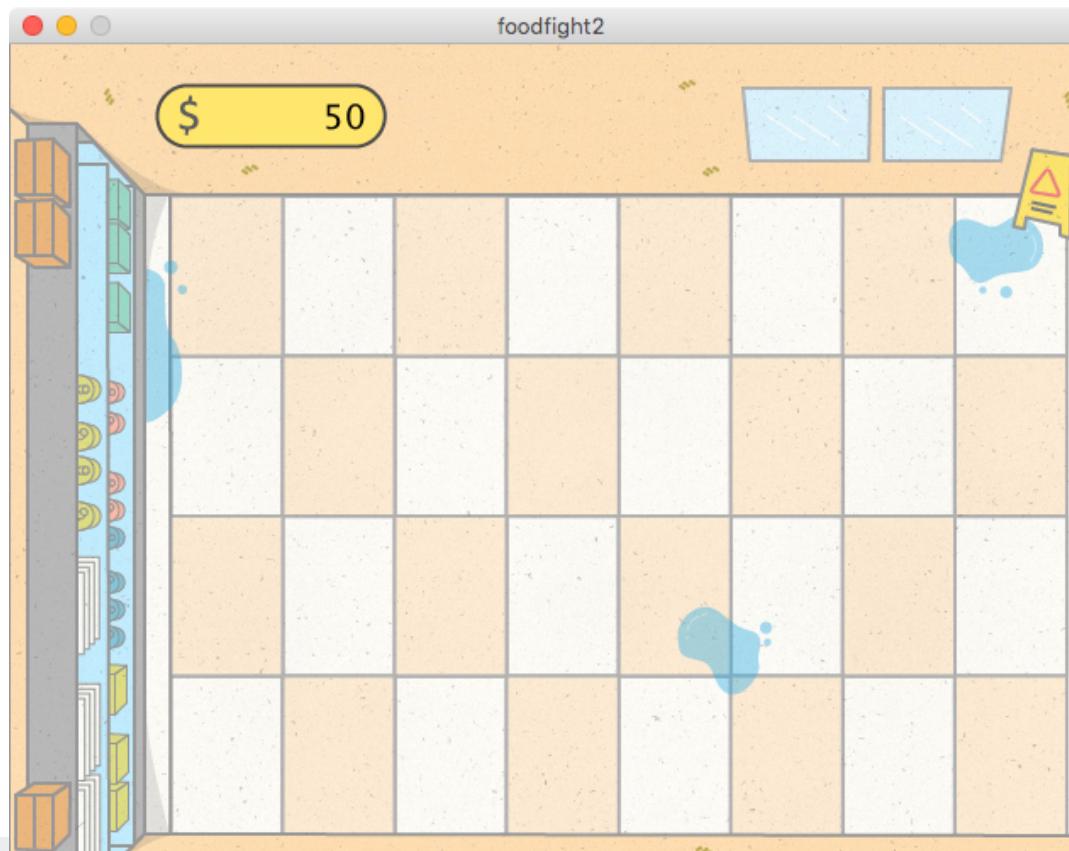
`fill(0);  
text( 2015-myAge , 210, 50); // birth year , x, y`

# 顯示圖片 與 文字

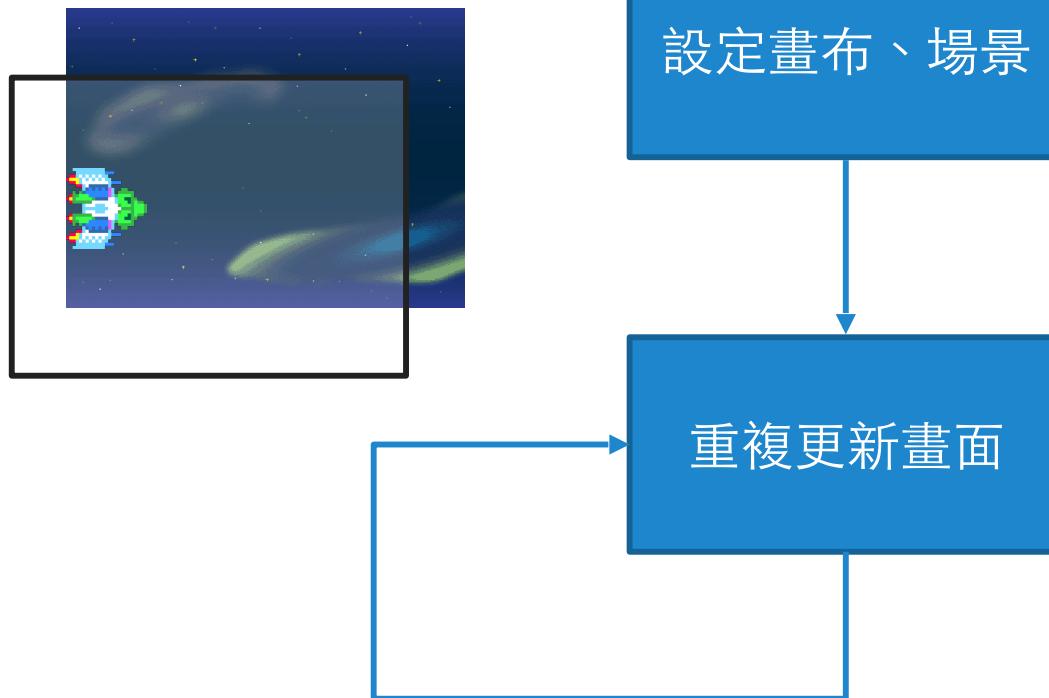
```
PImage bg; // declare a variable  
  
bg = loadImage("img/bg_play.jpg"); // initialize  
  
image(bg,0,0); // render image on the canvas  
  
// render text on the canvas  
  
textSize(20);  
  
textAlign(CENTER);  
  
fill(0); // fill color 0-255: grey scale  
  
text("Hello", 210,50);
```

# 小練習

- 使用 wallet 記錄口袋裡的錢，並顯示在畫面上



# 動畫原理



```
void setup(){  
}
```

```
void draw(){  
}
```

Function block

# 更新變數內容

```
wallet = wallet + 50;  
  
// or wallet += 50;  
  
// compound assignment operator  
  
// wallet++; -> wallet = wallet + 1;  
  
// Increment operator
```

# 加入水果按鈕 (objects)

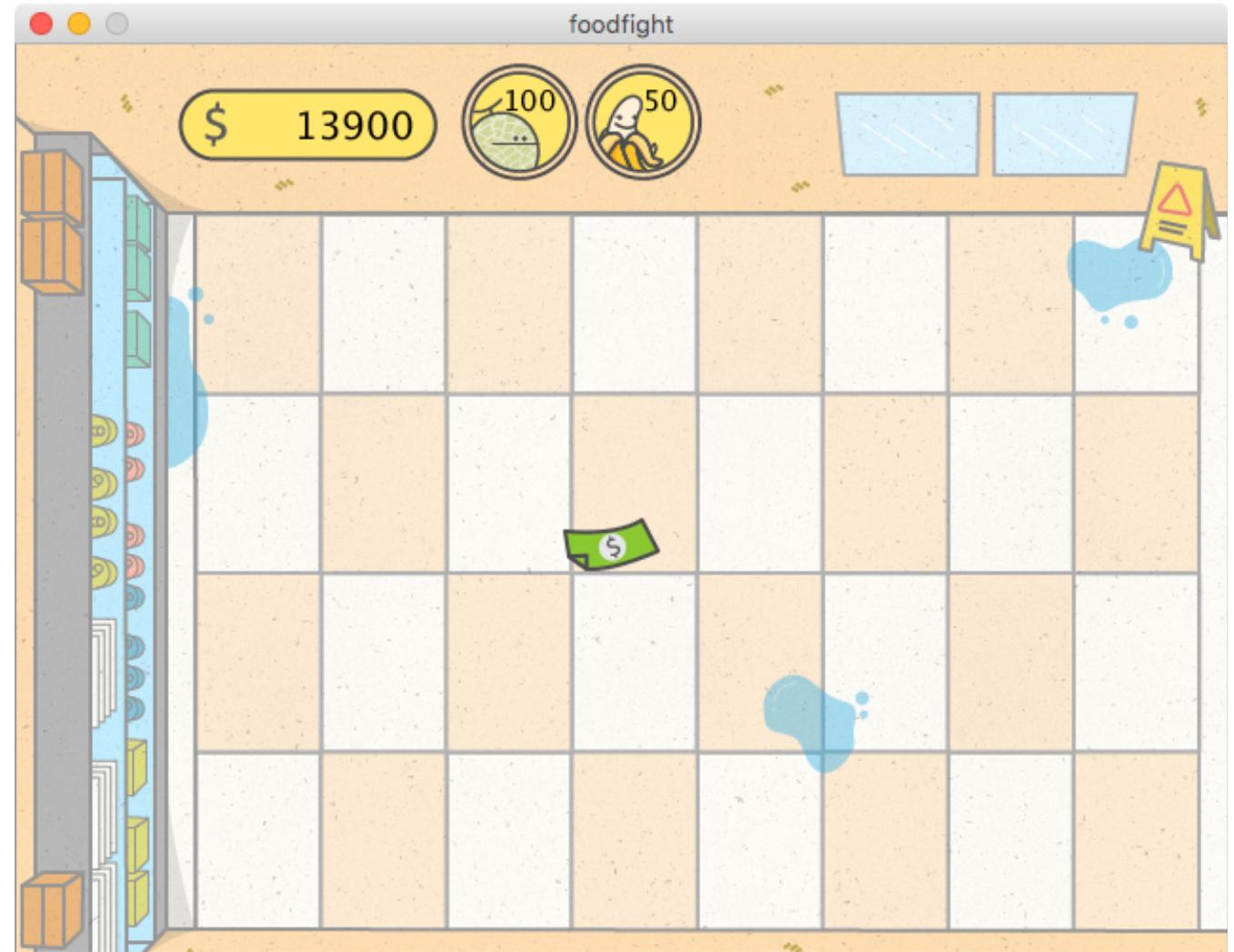
```
Button hami;  
  
void setup(){  
    hami = new Button("hami_btn", 235,10); // "banana_btn"  
}  
  
void draw(){  
    hami.display(); // method  
    textSize(15);  
    fill (0);  
    text("50", hami.x + 50 , hami.y +25) ; // properties  
}
```

# 鈔票飛上天

```
// 掉出螢幕外後，間隔ms  
// 再掉落下一張鈔票
```

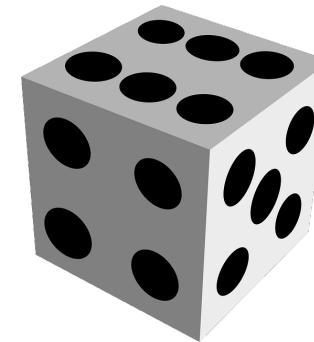
```
Money(int ms)
```

```
// 掉落動畫  
drop();
```



# 亂數 (Random number)

□ `random(n)` // n=6: 0~5.999999999



□ `int dice = floor(random(6));` // 0,1,2,3,4,5  
    無條件捨去

# 事件偵測

```
void keyPressed() {  
    money.x = random(width);  
}  
  
void mousePressed() {  
    money.y = random(height);  
}
```

# 牛奶不用錢：隨機放置毒牛奶在 0~3 列

// 工具程式庫

Util.getXbyCol(int)

例：Util.getXbyCol(7)

Util.getYbyRow(int)

Milk(int x, int y)

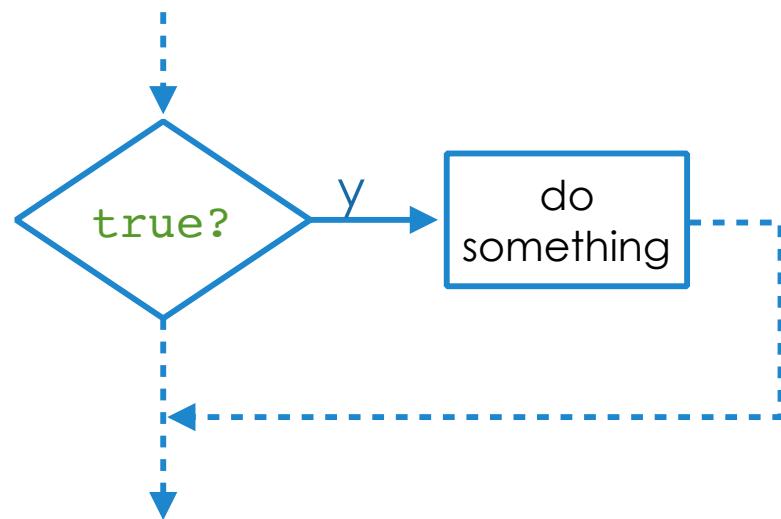
// 移動

move()



# 電腦判斷力

```
if (boolean_expression) {  
    // do something...  
}
```



- Use boolean operators
  - $a > b$
  - $a < b$
  - $a == b$
  - $a <= b$
  - $a >= b$
- $a != b$
- $a > b \&& b > c$
- $a > b || b > c$

# 揍毒奶

```
void keyPressed() {  
    if (key == 'h') {  
        m.hurt();  
    }  
}
```

# 超市淪陷

```
void draw( ){

    // ...

    if (m.x < 0){

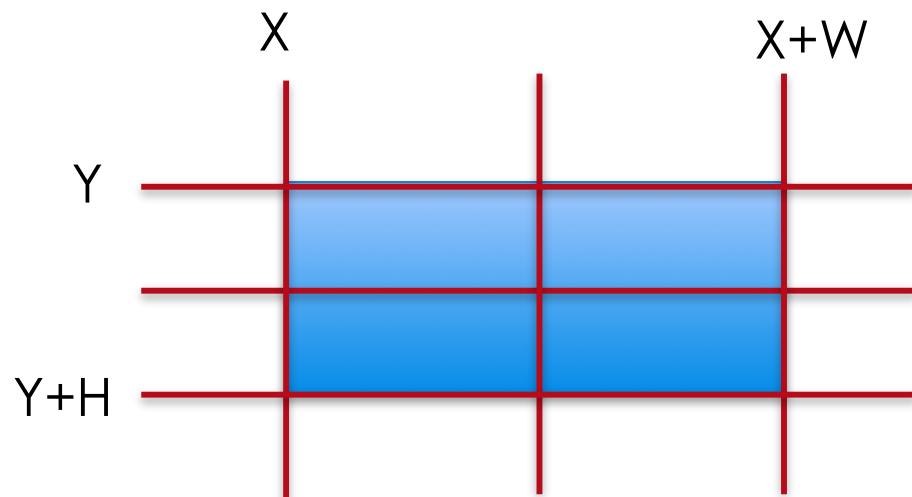
        image(bgOver,0,0); // "img/bg_lose.jpg"

    }

}
```

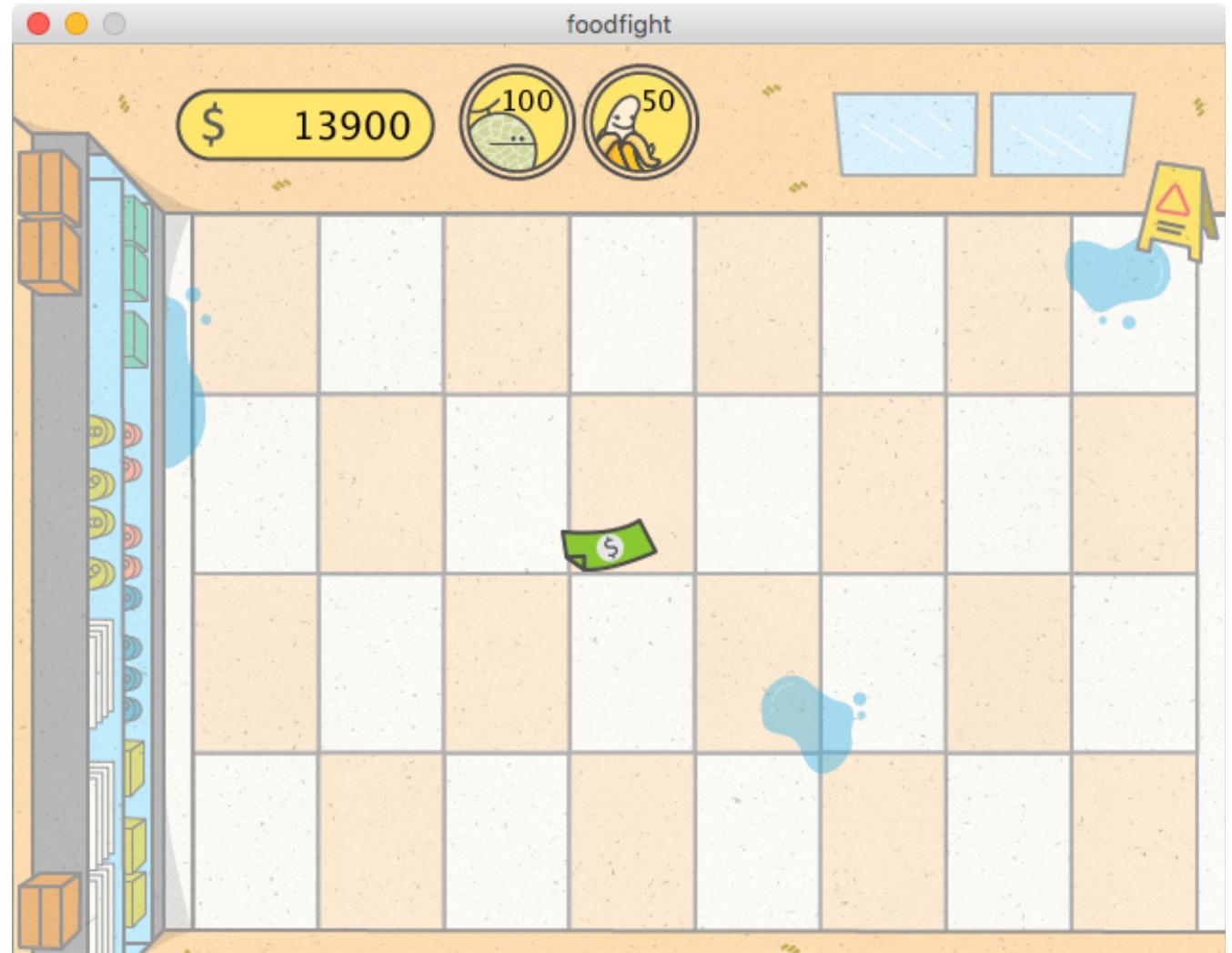
# 偵測滑鼠座標在矩形範圍中

```
if (mouseX > hami.x && mouseY > hami.y &&  
    mouseX < hami.x+hami.w && mouseY < hami.y+hami.h){  
    println("hami");  
}
```



# 撿一張鈔票 +50

```
// 掉出螢幕外後，間隔ms  
// 再掉落下一張鈔票  
  
Money(int ms)  
  
  
// 重新選擇掉落點  
restart();
```



# 常數 (Constants)

- 用途：儲存固定不變的數值
  - 初始化後，內容無法再做更動
  - 宣告方式：datatype 前加 **final**  
命名習慣：大寫字母加底線分格
- ```
final float MILES_KM_CONVERSION_VALUE = 1.61;
final float PI = 3.14;
final int NBA_FOUL_LIMIT = 6;
final int NBA_QUARTER_TIME_LENGTH = 12;
final int NBA_SHOT_CLOCK = 24;
final boolean DEBUG = true;
```

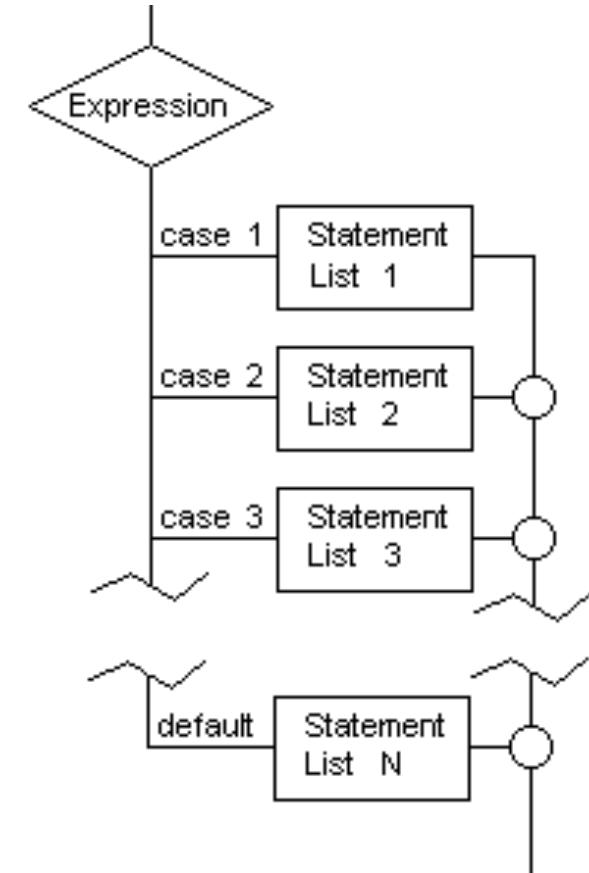
# 切換水果按鈕

```
// on Top          // in draw()  
  
final int BANANA = 1;      if (currFruit == BANANA){  
  
final int HAMI = 2;        hami.on = false;  
  
int currFruit;            banana.on = true;  
  
}else{  
  
// on setup()       hami.on = true;  
  
currFruit = HAMI;        banana.on = false;  
  
}  

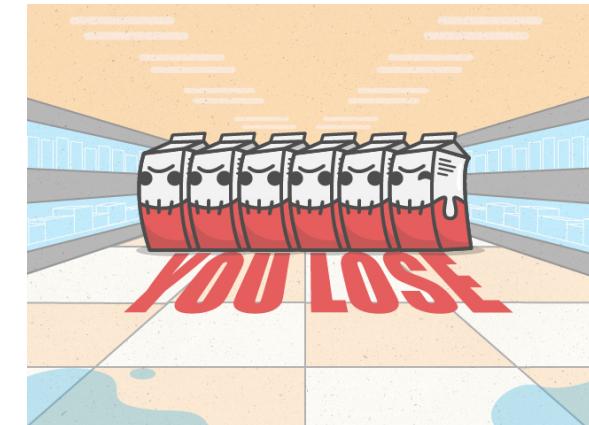
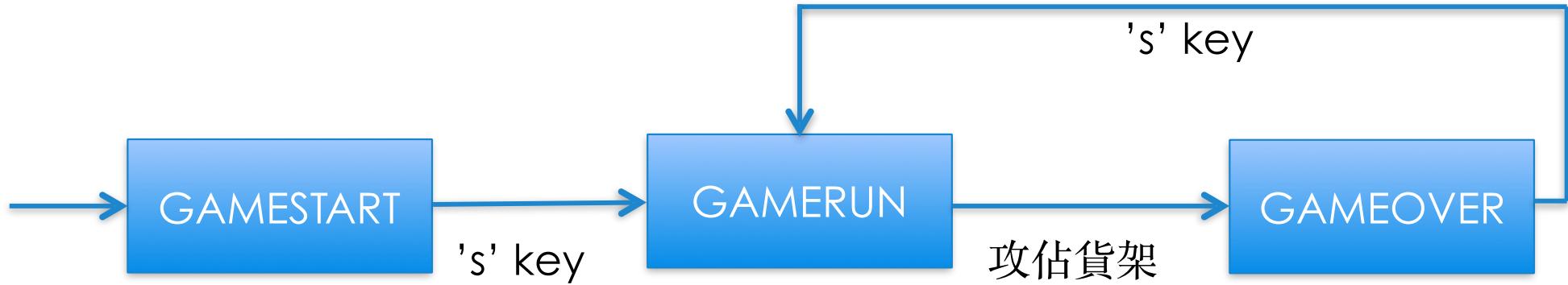
```

# switch statements

```
int or char  
switch ( expression ) {  
  
if  
    case cond1:  
        do something...;  
        break;  
  
else if  
    case cond2:  
        do something...;  
        break;  
  
else  
    default:  
        do something...;  
        break;  
}
```



# 流程控制



```

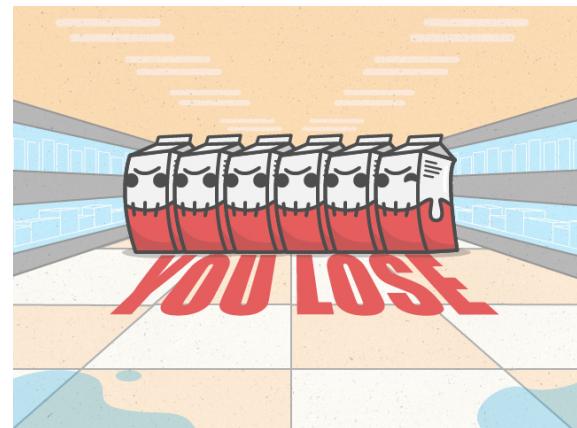
final int GAMESTART=0;
final int GAMERUN=1;
final int GAMEOVER=2;

int gameState;

void setup(){
    gameState = GAMESTART;
}

void draw(){
    switch (gameState){
        case GAMESTART:
            // do something
            break;
        case GAMERUN:
            // do something
            break;
        case GAMEOVER:
            // do something
            break;
    }
}

```



# FruitMgr

```
// on Top  
  
FruitMgr fruitMgr;  
  
// in setup()  
  
fruitMgr = new FruitMgr(false); // false: 不射子彈  
  
// in draw()  
  
fruitMgr.display();
```

# 放置水果在方格裡

```
// in mousePressed()

if (Util.isXYInGrid(mouseX,mouseY)){

    int col = Util.getColbyX(mouseX);

    int row = Util.getRowbyY(mouseY);

    println(col,row);

    boolean ok = fruitMgr.add(currFruit, col, row);

    // 新增成功 ok = true, 新增失敗 ok = false

}
```

# 水果新樂園：賺夠錢才能買水果

哈密瓜 100

香蕉 50



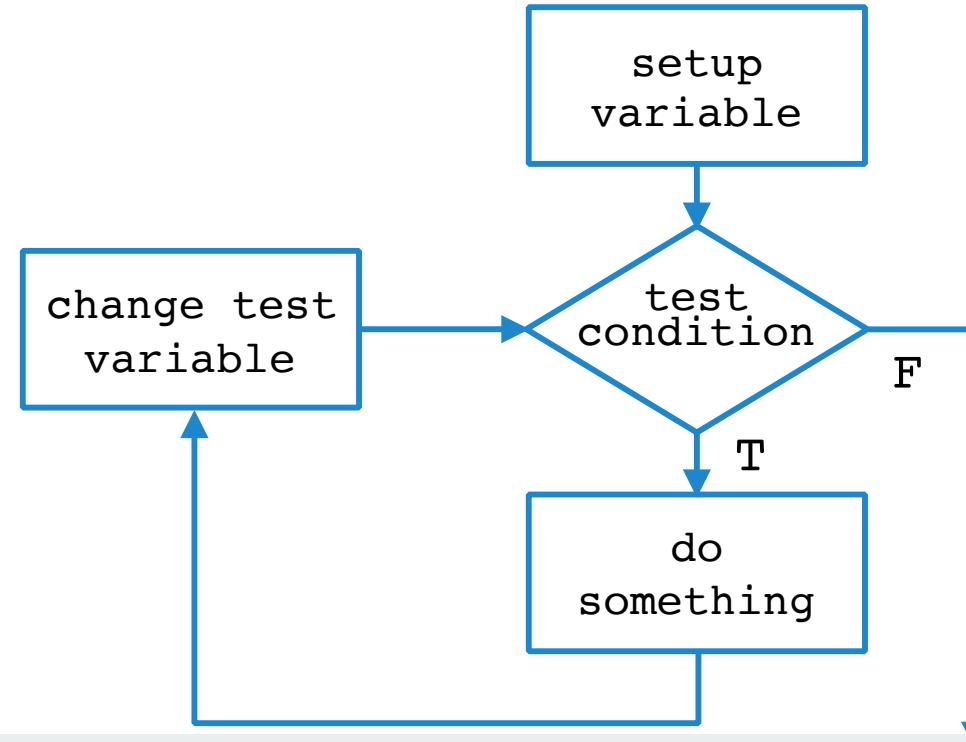
# 迴圈 (Loop)

```
for (setup variable; test condition; change test variable)  
{
```

```
    do something;
```

```
}
```

```
// in keyPressed()  
for (int i=0; i<7; i++){  
    // 把水果放進格子裡  
    fruitMgr.add(HAMI, i, 1);  
}
```



# Timer

```
// on Top

Timer t;

// in setup()

t = new Timer(1000); // 1000 ms = 1 second

// in draw()

if (t.isEnd()){    // 判斷計時是否結束

    fruitMgr.enableFire = true;

}
```

# 陣列 (ArrayList)

```
// on Top

ArrayList<Milk> milks; // declare

// in setup()

milks = new ArrayList<Milk>(); // initialize

for (int i=0; i < 4; i++){

    int x = Util.getXbyCol(7);

    int y = Util.getYbyRow(i);

    milks.add(new Milk(x,y)); // 加入陣列

}
```

# 陣列 (ArrayList)

```
for (int i=0; i < milks.size(); i++){      // in draw()

    Milk m = milks.get(i); // 以編號取出陣列裡的物件

    m.move();

    if (fruitMgr.isHitFruit(m.x,m.y,m.w,m.h)) {

        println("hit fruit");

    }

    if (fruitMgr.isHitBullet(m.x,m.y,m.w,m.h)) {

        m.hurt();

        if (m.isDead()){

            milks.remove(i);

        }

    }

}
```

# 進擊的毒奶

- 總共10罐牛奶
- 每隔2秒出現1罐
- 隨機選0~3列出現
- 初始固定在7th行



# Hint : 你可能需要以下變數

```
// on top  
  
int zombieNum; //毒奶總數  
  
int zombieCount; // 計數器：目前數量  
  
int zombieInterval; // 出現下一個毒奶的間隔時間  
  
Timer zombieTimer; // 計時器  
  
// in setup()  
  
zombieInterval = 2000; //ms  
  
zombieNum = 10;  
  
zombieCount = 0;  
  
zombieTimer = new Timer(zombieInterval);
```

# 宣告／呼叫自訂函式

return type      function name



```
void sayHi() {  
    println("Hi");  
}  
  
sayHi(); // Hi  
sayHi(); // Hi  
sayHi(); // Hi
```

# 參數傳遞

```
return type      function name      parameter / argument  
↑               ↑                   ↑  
void sayHi(String name) {  
  
    println("Hi " + name);  
  
}  
  
sayHi("Jones"); // Hi Jones  
  
sayHi("Alice"); // Hi Alice  
  
sayHi("Joe");  // Hi Joe
```

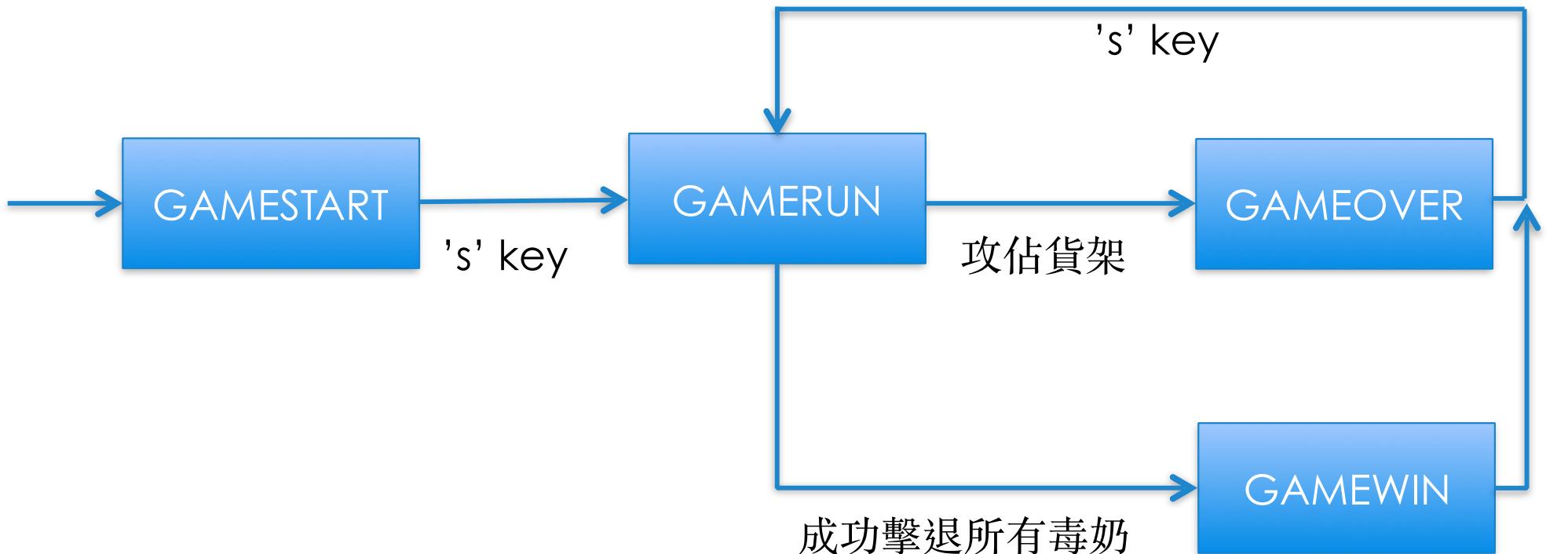
# 回傳值

```
return type    function name    parameter / argument  
↑             ↑                  ↑  
int square(int num) {  
  
    return(num * num );  
  
}  
  
int area = square(5); // call function  
  
//returns the int value 25
```

# 函數 (function)

```
void restart(){ // 遊戲重新開始  
    wallet = 0; // assignment operator  
    currFruit = BANANA;  
    zombieCount = 0;  
    money = new Money(3000); // delay 3000ms  
    fruitMgr = new FruitMgr(true);  
    zombieTimer = new Timer(zombieInterval);  
    milks = new ArrayList<Milk>();  
}
```

# 抵制有效：加入成功條件



\*請完成 boolean isWin()

zombieInterval變短  
zombieNum增加