iOS 應用程式開發

Final Poject

海綿寶寶~跳！跳！跳！

第 11 組

目錄

◆ 遊戲簡介

◆ 遊戲規則介紹

◆ 主要功能

◆ 問題與解決方式

一、遊戲簡介

讓海綿寶寶能突破各種障礙生存下去。

二、遊戲規則介紹

1. 一回合三條命。

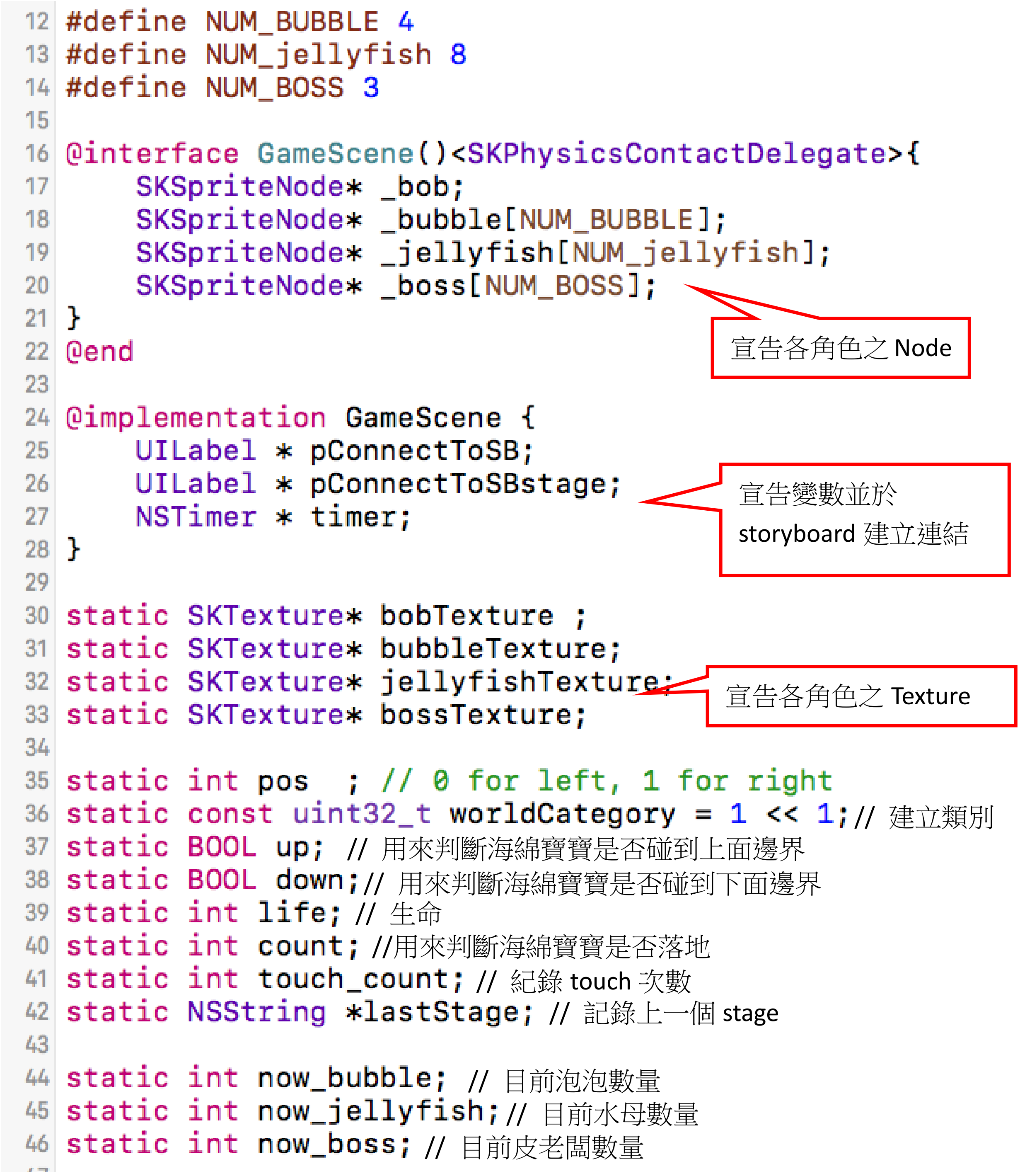
2. 點擊螢幕可使海綿寶寶在畫面中彈跳。

3. 吃一顆泡泡增加一命，撞到水母扣一命，撞到皮老闆則直接死亡。

4. 總共有 5 個 Level，每 15 秒進入下一個階段。

三、主要功能

- 變數宣告



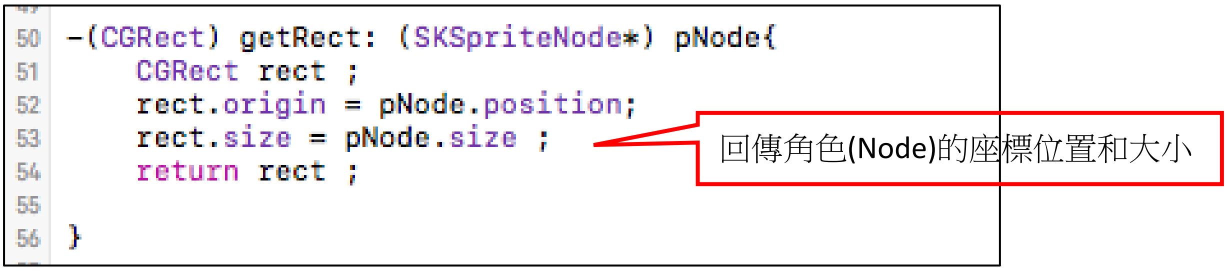
```
#define NUM_BUBBLE 4
#define NUM_jellyfish 8
#define NUM_BOSS 3

@interface GameScene()<SKPhysicsContactDelegate>{
    SKSpriteNode* _bob;
    SKSpriteNode* _bubble[NUM_BUBBLE];
    SKSpriteNode* _jellyfish[NUM_jellyfish];
    SKSpriteNode* _boss[NUM_BOSS];
}
@end

@implementation GameScene {
    UILabel * pConnectToSB;
    UILabel * pConnectToSBstage;
    NSTimer * timer;
}

static SKTexture* bobTexture ;
static SKTexture* bubbleTexture;
static SKTexture* jellyfishTexture;
static SKTexture* bossTexture;

static int pos  ; // 0 for left, 1 for right
static const uint32_t worldCategory = 1 << 1;// 建立類別
static BOOL up; // 用來判斷海綿寶寶是否碰到上面邊界
static BOOL down;// 用來判斷海綿寶寶是否碰到下面邊界
static int life; // 生命
static int count; //用來判斷海綿寶寶是否落地
static int touch_count; // 紀錄 touch 次數
static NSString *lastStage; // 記錄上一個 stage

static int now_bubble; // 目前泡泡數量
static int now_jellyfish;// 目前水母數量
static int now_boss; // 目前皮老闆數量
```

宣告各角色之 Node

宣告變數並於 storyboard 建立連結

宣告各角色之 Texture

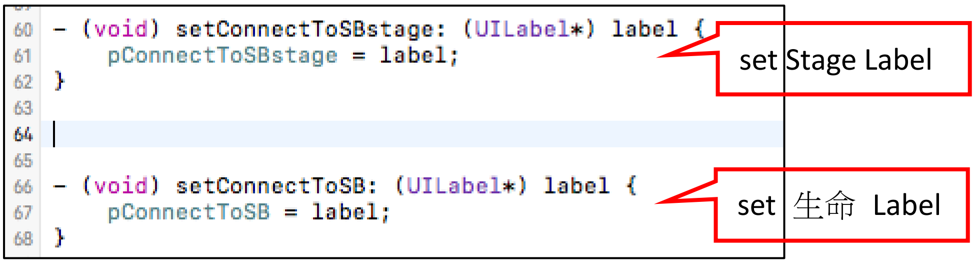
- getRect



```
-(CGRect) getRect: (SKSpriteNode*) pNode{
    CGRect rect ;
    rect.origin = pNode.position;
    rect.size = pNode.size ;
    return rect ;

}
```

回傳角色(Node)的座標位置和大小

- SetLabel（傳 LABEL 給 Storyboard 用的）

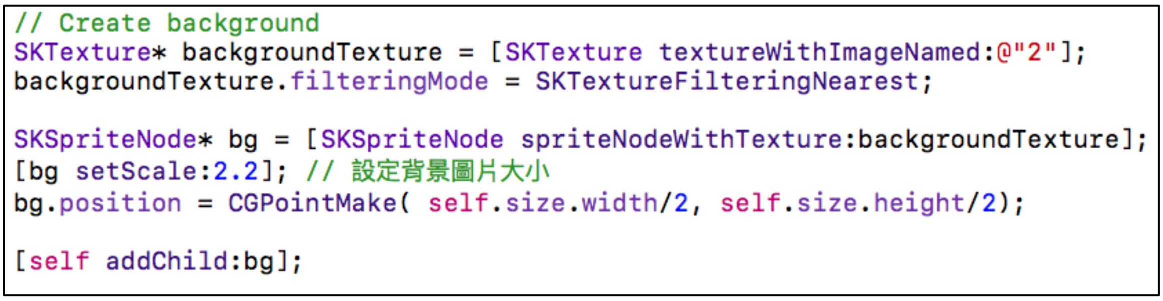


```
- (void) setConnectToSBstage: (UILabel*) label {
    pConnectToSBstage = label;
}



- (void) setConnectToSB: (UILabel*) label {
    pConnectToSB = label;
}
```

set Stage Label

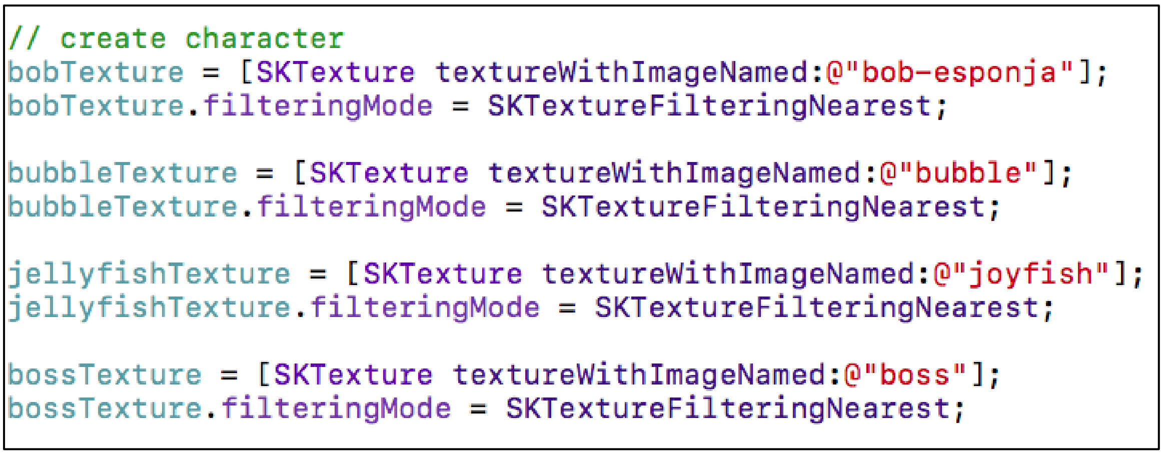
set 生命 Label

- 利用 SKTexture 建立背景並 assign 給 bg（SKSriteNode）



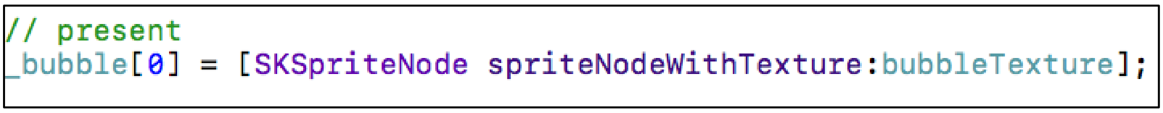
```
// Create background
SKTexture* backgroundTexture = [SKTexture textureWithImageNamed:@"2"];
backgroundTexture.filteringMode = SKTextureFilteringNearest;

SKSpriteNode* bg = [SKSpriteNode spriteNodeWithTexture:backgroundTexture];
[bg setScale:2.2]; // 設定背景圖片大小
bg.position = CGPointMake( self.size.width/2, self.size.height/2);

[self addChild:bg];
```

- 利用 SKTexture 建立各角色(之後要 assign 給物件)



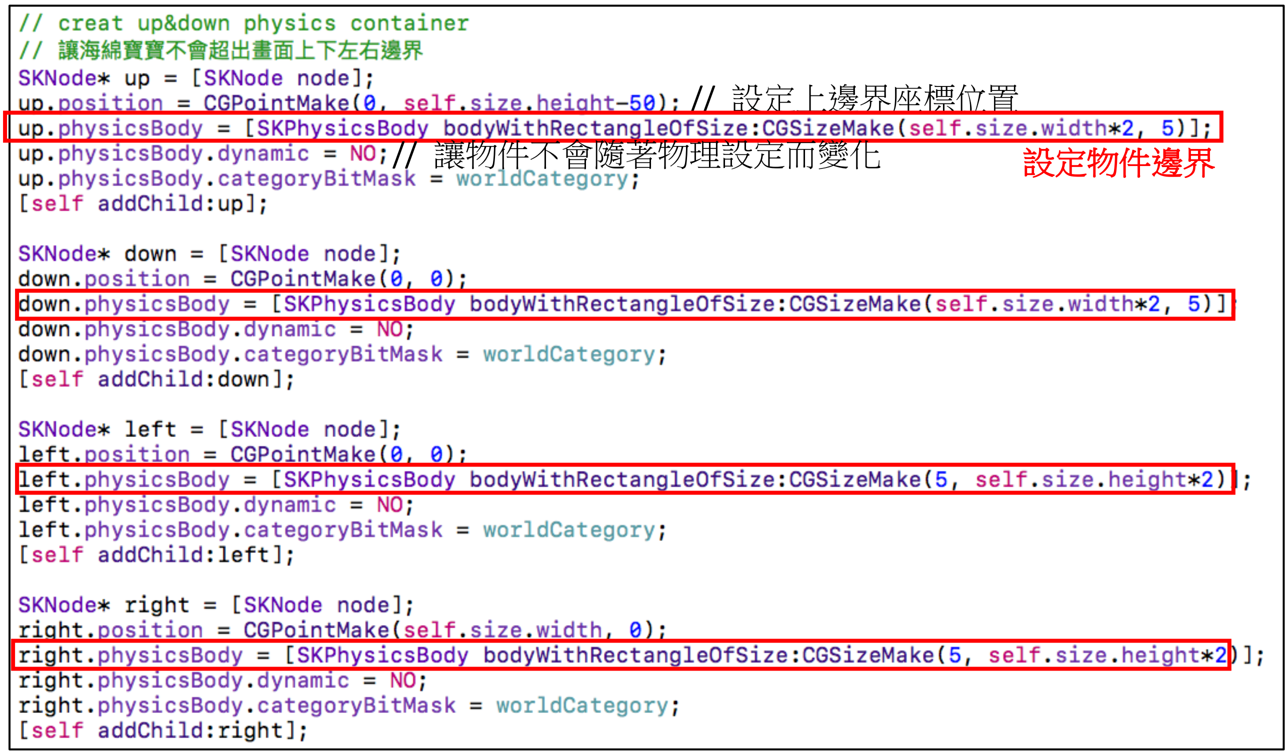
```
// create character
bobTexture = [SKTexture textureWithImageNamed:@"bob-esponja"];
bobTexture.filteringMode = SKTextureFilteringNearest;

bubbleTexture = [SKTexture textureWithImageNamed:@"bubble"];
bubbleTexture.filteringMode = SKTextureFilteringNearest;

jellyfishTexture = [SKTexture textureWithImageNamed:@"joyfish"];
jellyfishTexture.filteringMode = SKTextureFilteringNearest;

bossTexture = [SKTexture textureWithImageNamed:@"boss"];
bossTexture.filteringMode = SKTextureFilteringNearest;
```

- Assign 給 bubble (SKSriteNode)讓泡泡顯示於畫面上



```
// present
_bubble[0] = [SKSpriteNode spriteNodeWithTexture:bubbleTexture];
```

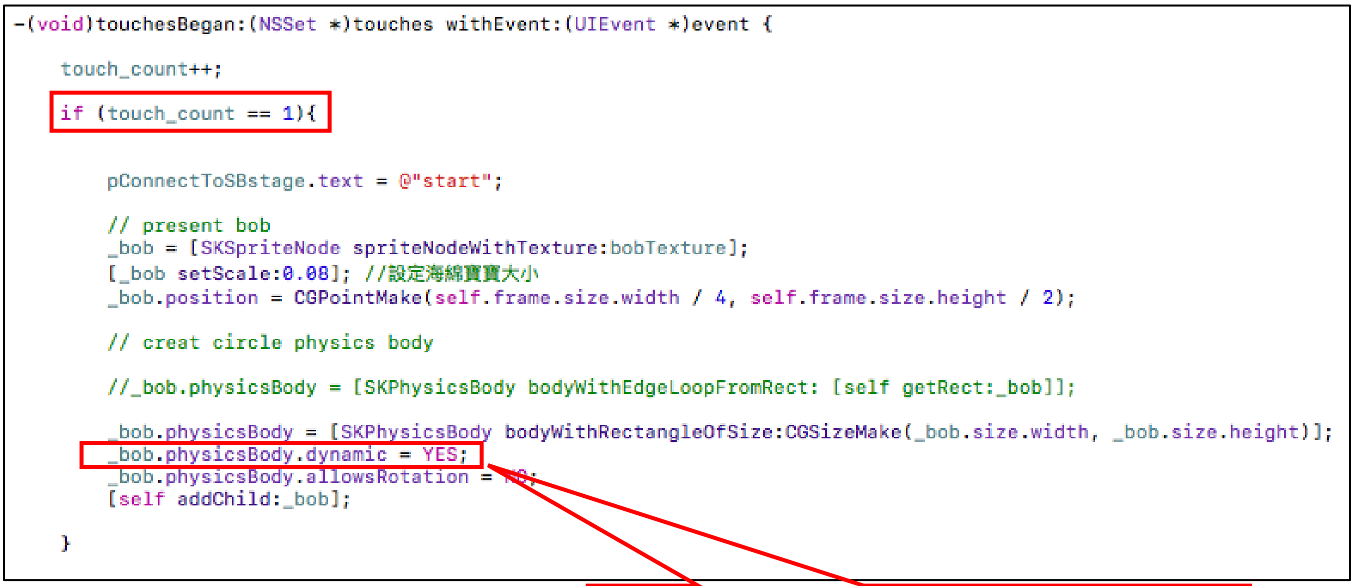
- 設定畫面上、下、左、右邊界，防止海綿寶寶超出畫面



```
// creat up&down physics container
// 讓海綿寶寶不會超出畫面上下左右邊界
SKNode* up = [SKNode node];
up.position = CGPointMake(0, self.size.height-50); // 設定上邊界座標位置
up.physicsBody = [SKPhysicsBody bodyWithRectangleOfSize:CGSizeMake(self.size.width*2, 5)];
up.physicsBody.dynamic = NO;// 讓物件不會隨著物理設定而變化
up.physicsBody.categoryBitMask = worldCategory;
[self addChild:up];

SKNode* down = [SKNode node];
down.position = CGPointMake(0, 0);
down.physicsBody = [SKPhysicsBody bodyWithRectangleOfSize:CGSizeMake(self.size.width*2, 5)];
down.physicsBody.dynamic = NO;
down.physicsBody.categoryBitMask = worldCategory;
[self addChild:down];

SKNode* left = [SKNode node];
left.position = CGPointMake(0, 0);
left.physicsBody = [SKPhysicsBody bodyWithRectangleOfSize:CGSizeMake(5, self.size.height*2)];
left.physicsBody.dynamic = NO;
left.physicsBody.categoryBitMask = worldCategory;
[self addChild:left];

SKNode* right = [SKNode node];
right.position = CGPointMake(self.size.width, 0);
right.physicsBody = [SKPhysicsBody bodyWithRectangleOfSize:CGSizeMake(5, self.size.height*2)];
right.physicsBody.dynamic = NO;
right.physicsBody.categoryBitMask = worldCategory;
[self addChild:right];
```

設定物件邊界

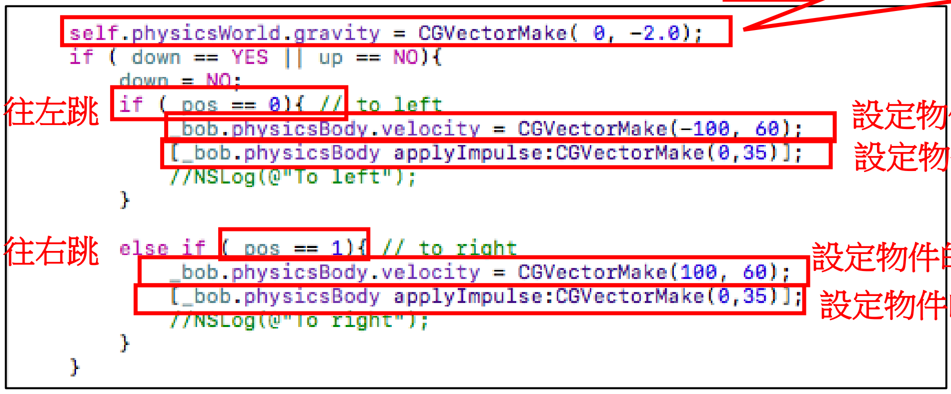
- TouchesBegan（觸碰螢幕會做的事）



```
-(void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event {

    touch_count++;

    if (touch_count == 1){

        pConnectToSBstage.text = @"start";

        // present bob
        _bob = [SKSpriteNode spriteNodeWithTexture:bobTexture];
        [_bob setScale:0.08]; //設定海綿寶寶大小
        _bob.position = CGPointMake(self.frame.size.width / 4, self.frame.size.height / 2);

        // creat circle physics body

        //_bob.physicsBody = [SKPhysicsBody bodyWithEdgeLoopFromRect: [self getRect:_bob]];

        _bob.physicsBody = [SKPhysicsBody bodyWithRectangleOfSize:CGSizeMake(_bob.size.width, _bob.size.height)];
        _bob.physicsBody.dynamic = YES;
        _bob.physicsBody.allowsRotation = NO;
        [self addChild:_bob];

    }
```

觸碰螢幕的次數為 1，
Dynamic 設為 YES，讓物件隨著物理設定變化
海綿寶寶才能跳

設定整個世界的重力，
物件 dynamic 若為 yes 會受此影響



```
    self.physicsWorld.gravity = CGVectorMake( 0, -2.0);
    if ( down == YES || up == NO){
        down = NO;
        if ( pos == 0){ // to left
            _bob.physicsBody.velocity = CGVectorMake(-100, 60);
            [_bob.physicsBody applyImpulse:CGVectorMake(0,35)];
            //NSLog(@"To left");
        }

        else if ( pos == 1){ // to right
            _bob.physicsBody.velocity = CGVectorMake(100, 60);
            [_bob.physicsBody applyImpulse:CGVectorMake(0,35)];
            //NSLog(@"To right");
        }
    }
```

往左跳

設定物件的速度 **-100 為往左**

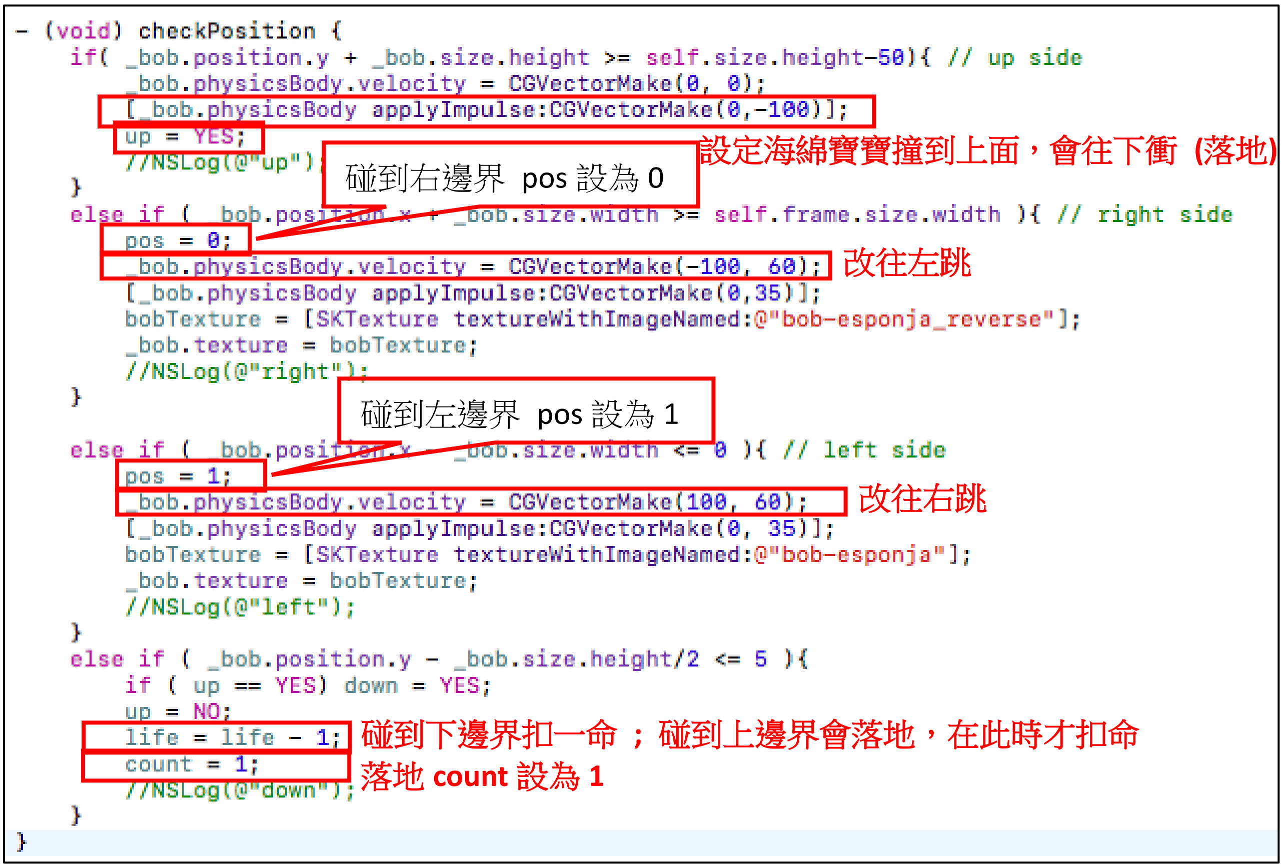
設定物件的衝力 **(往上跳)**

往右跳

設定物件的速度 **100 為往右**

設定物件的衝力

- check 海綿寶寶的位置



```
- (void) checkPosition {
    if( _bob.position.y + _bob.size.height >= self.size.height-50){ // up side
        _bob.physicsBody.velocity = CGVectorMake(0, 0);
        [_bob.physicsBody applyImpulse:CGVectorMake(0,-100)];
        up = YES;
        //NSLog(@"up");
    }
    else if ( _bob.position.x + _bob.size.width >= self.frame.size.width ){ // right side
        pos = 0;
        _bob.physicsBody.velocity = CGVectorMake(-100, 60);
        [_bob.physicsBody applyImpulse:CGVectorMake(0,35)];
        bobTexture = [SKTexture textureWithImageNamed:@"bob-esponja_reverse"];
        _bob.texture = bobTexture;
        //NSLog(@"right");
    }

    else if ( _bob.position.x - _bob.size.width <= 0 ){ // left side
        pos = 1;
        _bob.physicsBody.velocity = CGVectorMake(100, 60);
        [_bob.physicsBody applyImpulse:CGVectorMake(0, 35)];
        bobTexture = [SKTexture textureWithImageNamed:@"bob-esponja"];
        _bob.texture = bobTexture;
        //NSLog(@"left");
    }
    else if ( _bob.position.y - _bob.size.height/2 <= 5 ){
        if ( up == YES) down = YES;
        up = NO;
        life = life - 1;
        count = 1;
        //NSLog(@"down");
    }
}
```

設定海綿寶寶撞到上面，會往下衝 (落地)

碰到右邊界 pos 設為 0

改往左跳

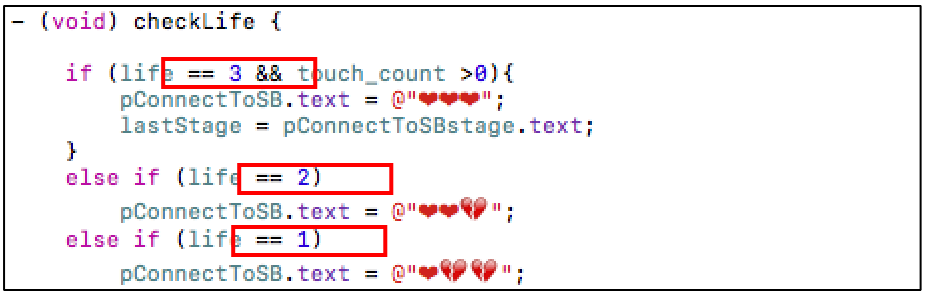
碰到左邊界 pos 設為 1

改往右跳

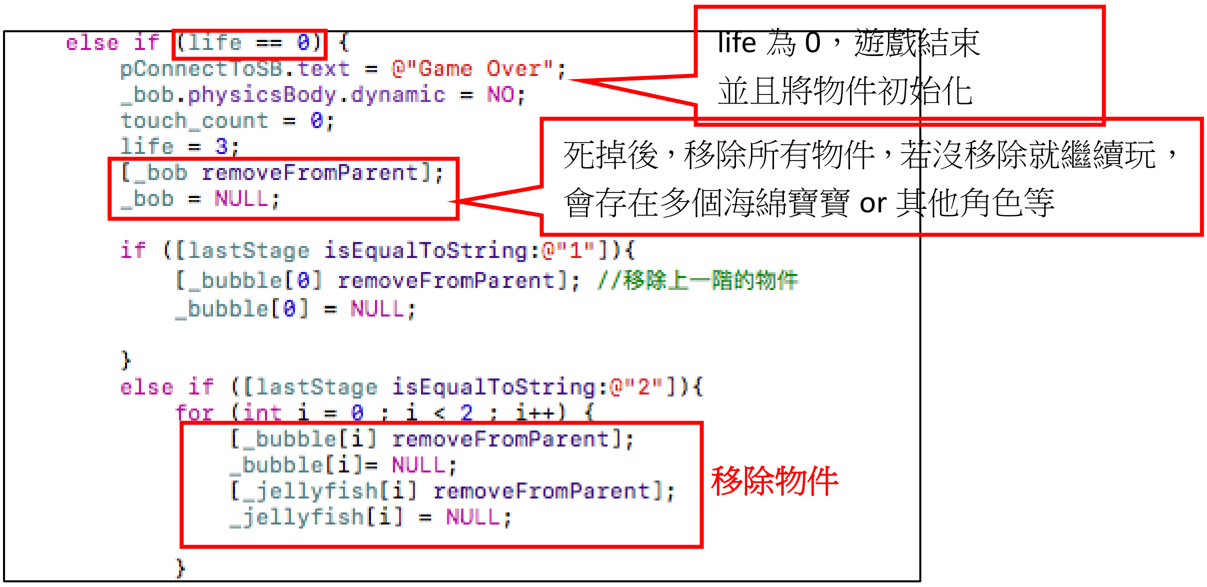
碰到下邊界扣一命 ； 碰到上邊界會落地，在此時才扣命

落地 count 設為 1

- check 海綿寶寶的生命



```
- (void) checkLife {

    if (life == 3 && touch_count >0){
        pConnectToSB.text = @"❤❤❤";
        lastStage = pConnectToSBstage.text;
    }
    else if (life == 2)
        pConnectToSB.text = @"❤❤💔";
    else if (life == 1)
        pConnectToSB.text = @"❤💔💔";
```



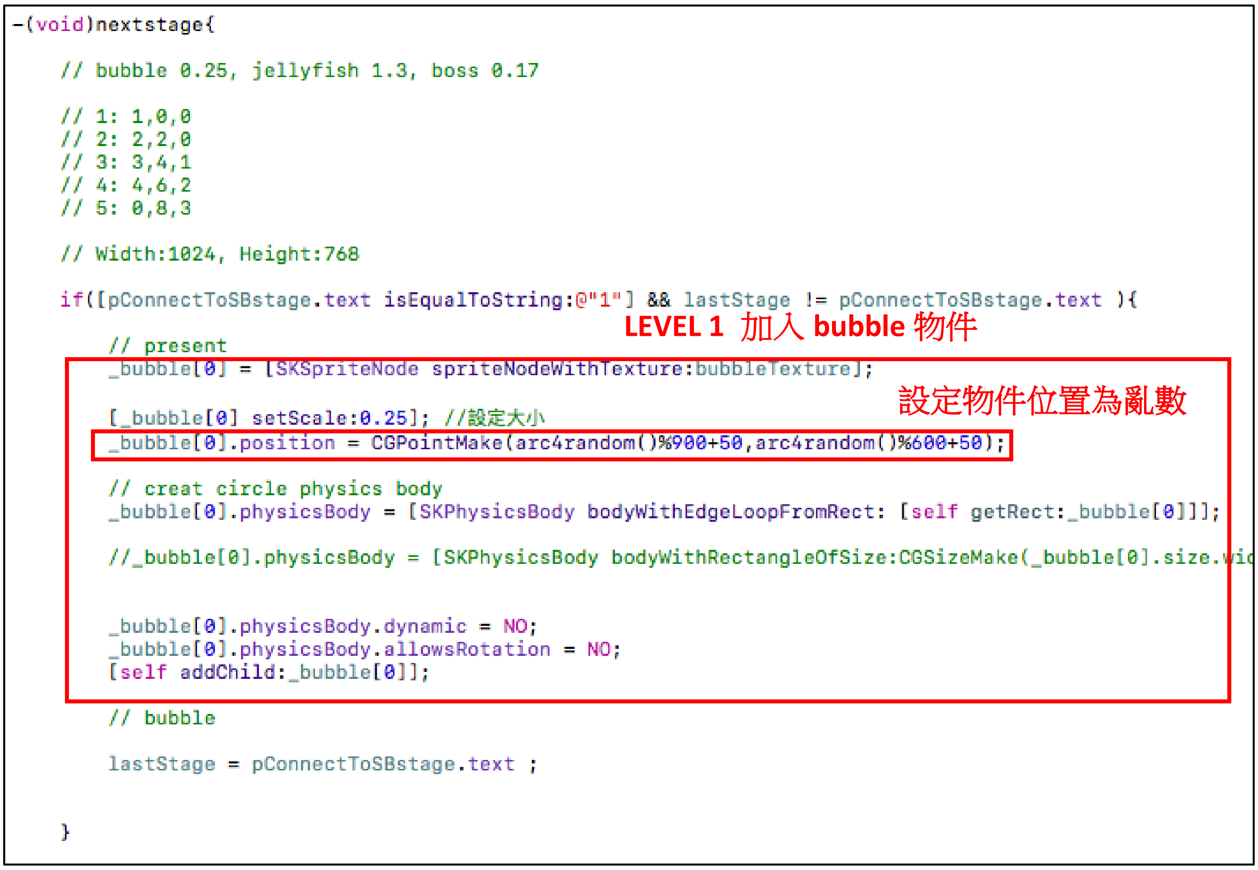
```
    else if (life == 0) {
        pConnectToSB.text = @"Game Over";
        _bob.physicsBody.dynamic = NO;
        touch_count = 0;
        life = 3;
        [_bob removeFromParent];
        _bob = NULL;

        if ([lastStage isEqualToString:@"1"]){
            [_bubble[0] removeFromParent]; //移除上一階的物件
            _bubble[0] = NULL;

        }
        else if ([lastStage isEqualToString:@"2"]){
            for (int i = 0 ; i < 2 ; i++) {
                [_bubble[i] removeFromParent];
                _bubble[i]= NULL;
                [_jellyfish[i] removeFromParent];
                _jellyfish[i] = NULL;
            }
```

life 為 0，遊戲結束
並且將物件初始化

死掉後，移除所有物件，若沒移除就繼續玩，
會存在多個海綿寶寶 or 其他角色等

移除物件

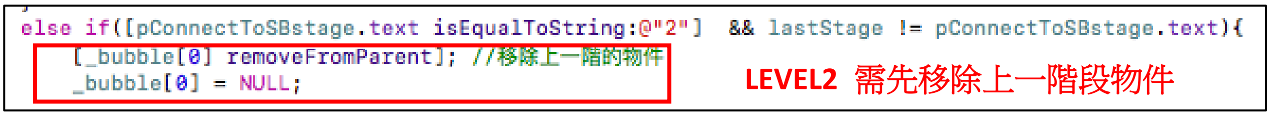
- 處理切換 Stage(level)之內容並清畫面



```
-(void)nextstage{

    // bubble 0.25, jellyfish 1.3, boss 0.17

    // 1: 1,0,0
    // 2: 2,2,0
    // 3: 3,4,1
    // 4: 4,6,2
    // 5: 0,8,3

    // Width:1024, Height:768

    if([pConnectToSBstage.text isEqualToString:@"1"] && lastStage != pConnectToSBstage.text ){

        // present
        _bubble[0] = [SKSpriteNode spriteNodeWithTexture:bubbleTexture];

        [_bubble[0] setScale:0.25]; //設定大小
        _bubble[0].position = CGPointMake(arc4random()%900+50,arc4random()%600+50);

        // creat circle physics body
        _bubble[0].physicsBody = [SKPhysicsBody bodyWithEdgeLoopFromRect: [self getRect:_bubble[0]]];

        //_bubble[0].physicsBody = [SKPhysicsBody bodyWithRectangleOfSize:CGSizeMake(_bubble[0].size.wid


        _bubble[0].physicsBody.dynamic = NO;
        _bubble[0].physicsBody.allowsRotation = NO;
        [self addChild:_bubble[0]];
        // bubble

        lastStage = pConnectToSBstage.text ;


    }
```

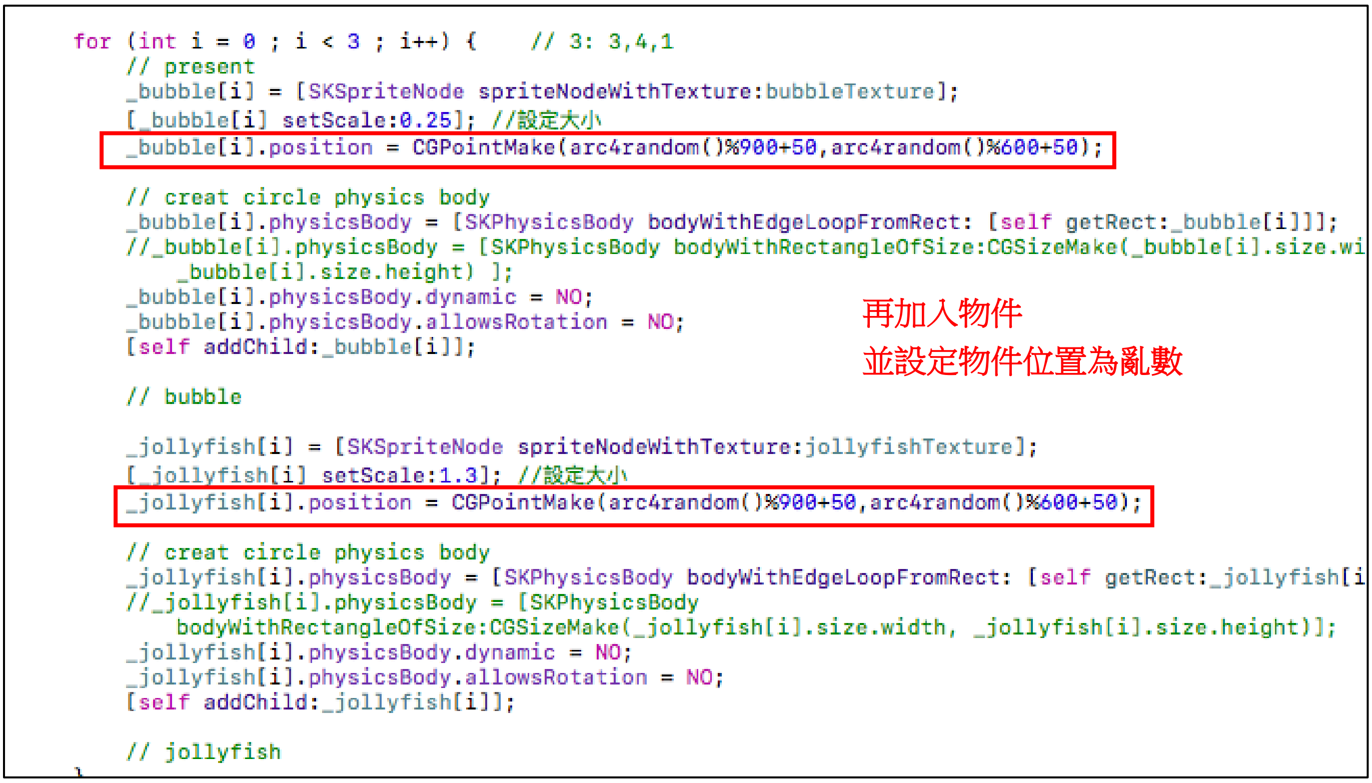
LEVEL 1 加入 bubble 物件

設定物件位置為亂數



```
else if([pConnectToSBstage.text isEqualToString:@"2"]  && lastStage != pConnectToSBstage.text){
    [_bubble[0] removeFromParent]; //移除上一階的物件
    _bubble[0] = NULL;
```

LEVEL2 需先移除上一階段物件

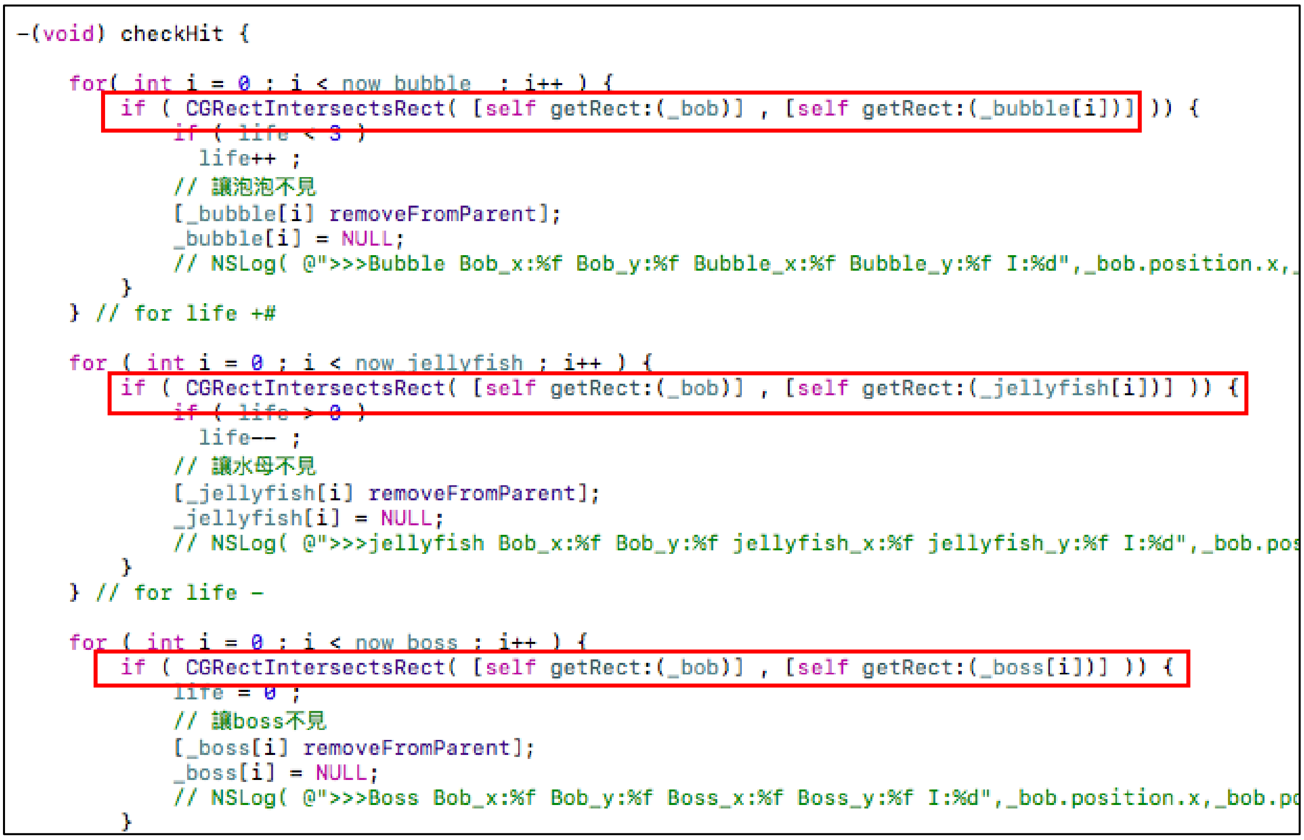


```
for (int i = 0 ; i < 3 ; i++) {    // 3: 3,4,1
    // present
    _bubble[i] = [SKSpriteNode spriteNodeWithTexture:bubbleTexture];
    [_bubble[i] setScale:0.25]; //設定大小
    _bubble[i].position = CGPointMake(arc4random()%900+50,arc4random()%600+50);

    // creat circle physics body
    _bubble[i].physicsBody = [SKPhysicsBody bodyWithEdgeLoopFromRect: [self getRect:_bubble[i]]];
    //_bubble[i].physicsBody = [SKPhysicsBody bodyWithRectangleOfSize:CGSizeMake(_bubble[i].size.wi
        _bubble[i].size.height) ];
    _bubble[i].physicsBody.dynamic = NO;
    _bubble[i].physicsBody.allowsRotation = NO;
    [self addChild:_bubble[i]];

    // bubble

    _jollyfish[i] = [SKSpriteNode spriteNodeWithTexture:jollyfishTexture];
    [_jollyfish[i] setScale:1.3]; //設定大小
    _jollyfish[i].position = CGPointMake(arc4random()%900+50,arc4random()%600+50);

    // creat circle physics body
    _jollyfish[i].physicsBody = [SKPhysicsBody bodyWithEdgeLoopFromRect: [self getRect:_jollyfish[i
    //_jollyfish[i].physicsBody = [SKPhysicsBody
        bodyWithRectangleOfSize:CGSizeMake(_jollyfish[i].size.width, _jollyfish[i].size.height)];
    _jollyfish[i].physicsBody.dynamic = NO;
    _jollyfish[i].physicsBody.allowsRotation = NO;
    [self addChild:_jollyfish[i]];

    // jollyfish
```

再加入物件

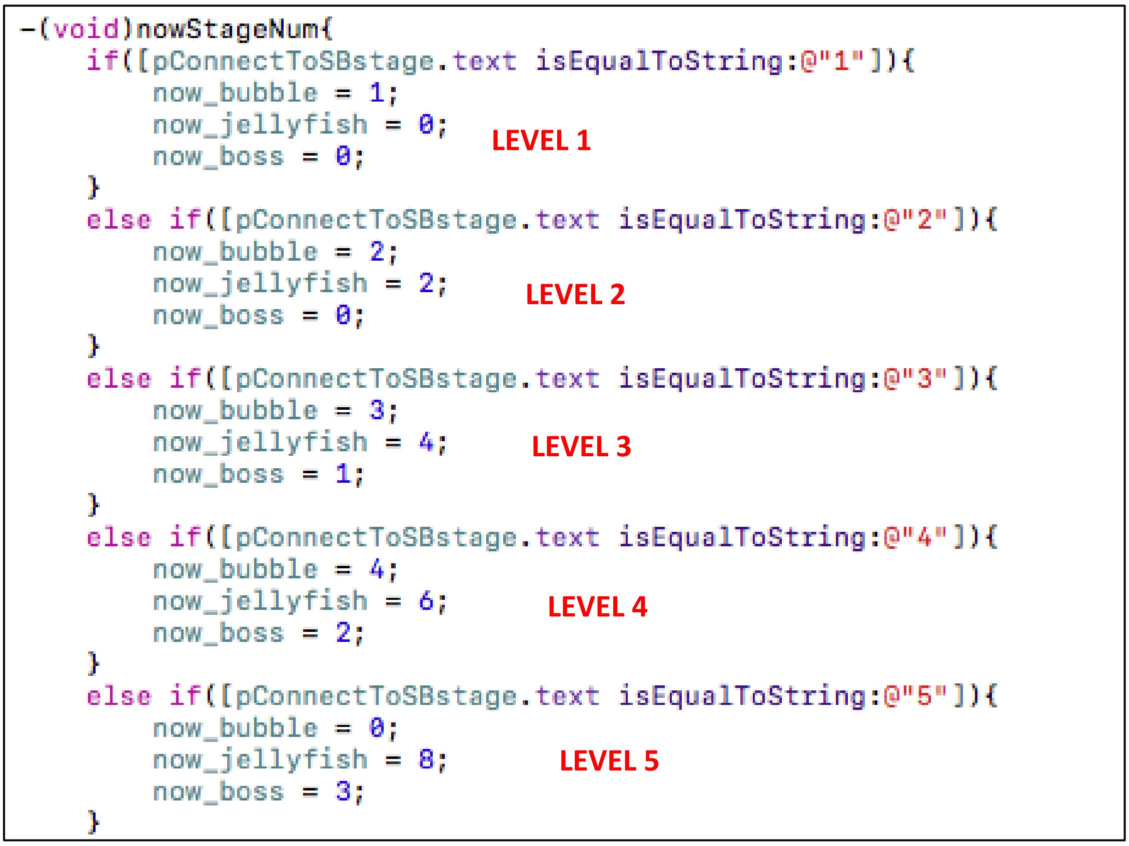
並設定物件位置為亂數

- 利用 CGRect 判斷物件是否碰撞到



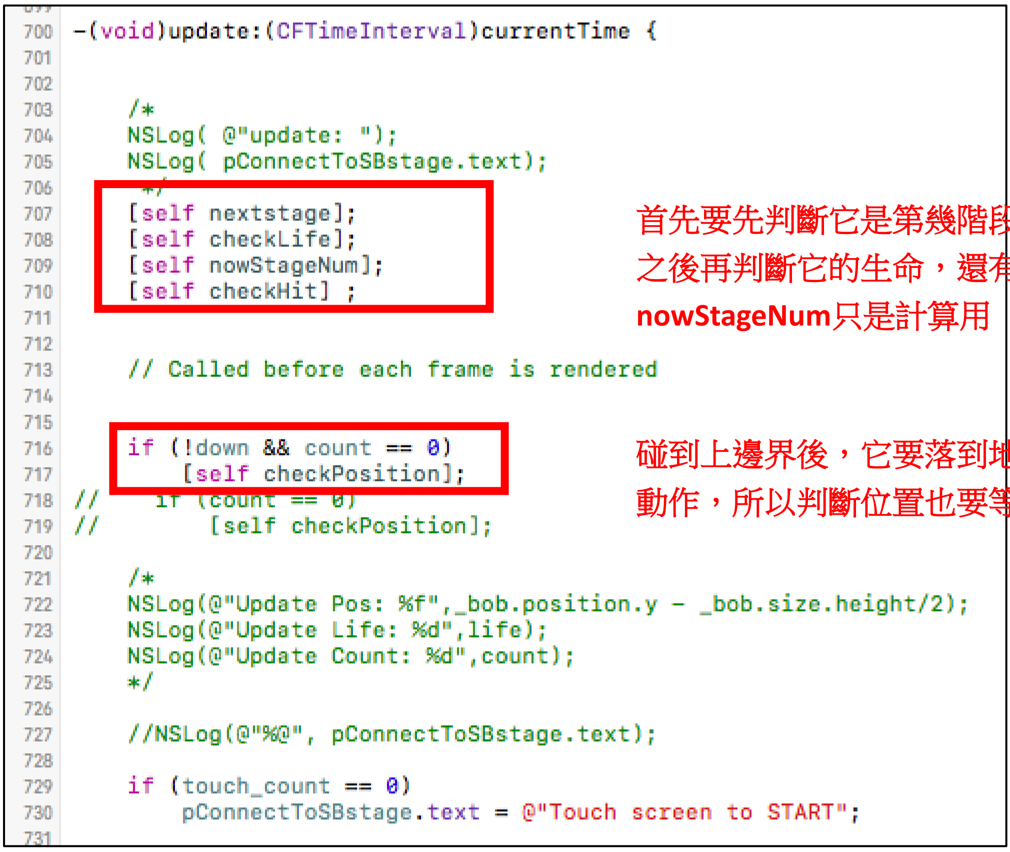
```
-(void) checkHit {

    for( int i = 0 ; i < now_bubble  ; i++ ) {
        if ( CGRectIntersectsRect( [self getRect:(_bobble)] , [self getRect:(_bubble[i])] )) {
            if ( life < 3 )
              life++ ;
            // 讓泡泡不見
            [_bubble[i] removeFromParent];
            _bubble[i] = NULL;
            // NSLog( @">>>Bubble Bob_x:%f Bob_y:%f Bubble_x:%f Bubble_y:%f I:%d",_bob.position.x,_
        }
    } // for life +#

    for ( int i = 0 ; i < now_jellyfish ; i++ ) {
        if ( CGRectIntersectsRect( [self getRect:(_bob)] , [self getRect:(_jellyfish[i])] )) {
            if ( life > 0 )
              life-- ;
            // 讓水母不見
            [_jellyfish[i] removeFromParent];
            _jellyfish[i] = NULL;
            // NSLog( @">>>jellyfish Bob_x:%f Bob_y:%f jellyfish_x:%f jellyfish_y:%f I:%d",_bob.pos
        }
    } // for life -

    for ( int i = 0 ; i < now_boss ; i++ ) {
        if ( CGRectIntersectsRect( [self getRect:(_bob)] , [self getRect:(_boss[i])] )) {
            life = 0 ;
            // 讓boss不見
            [_boss[i] removeFromParent];
            _boss[i] = NULL;
            // NSLog( @">>>Boss Bob_x:%f Bob_y:%f Boss_x:%f Boss_y:%f I:%d",_bob.position.x,_bob.po
        }
```

- 設定各階段怪物數量



```
-(void)nowStageNum{
    if([pConnectToSBstage.text isEqualToString:@"1"]){
        now_bubble = 1;
        now_jellyfish = 0;      LEVEL 1
        now_boss = 0;
    }
    else if([pConnectToSBstage.text isEqualToString:@"2"]){
        now_bubble = 2;
        now_jellyfish = 2;       LEVEL 2
        now_boss = 0;
    }
    else if([pConnectToSBstage.text isEqualToString:@"3"]){
        now_bubble = 3;
        now_jellyfish = 4;       LEVEL 3
        now_boss = 1;
    }
    else if([pConnectToSBstage.text isEqualToString:@"4"]){
        now_bubble = 4;
        now_jellyfish = 6;        LEVEL 4
        now_boss = 2;
    }
    else if([pConnectToSBstage.text isEqualToString:@"5"]){
        now_bubble = 0;
        now_jellyfish = 8;         LEVEL 5
        now_boss = 3;
    }
```

- Update



```
-(void)update:(CFTimeInterval)currentTime {


    /*
    NSLog( @"update: ");
    NSLog( pConnectToSBstage.text);
     */
    [self nextstage];
    [self checkLife];
    [self nowStageNum];
    [self checkHit] ;

    // Called before each frame is rendered

    if (!down && count == 0)
        [self checkPosition];
//    if (count == 0)
//        [self checkPosition];

    /*
    NSLog(@"Update Pos: %f",_bob.position.y - _bob.size.height/2);
    NSLog(@"Update Life: %d",life);
    NSLog(@"Update Count: %d",count);
    */

    //NSLog(@"%@", pConnectToSBstage.text);

    if (touch_count == 0)
        pConnectToSBstage.text = @"Touch screen to START";
```

首先要先判斷它是第幾階段，這最重要
之後再判斷它的生命，還有是否有碰撞
nowStageNum只是計算用

碰到上邊界後，它要落到地面才可以再次做
動作，所以判斷位置也要等落地之後才判斷

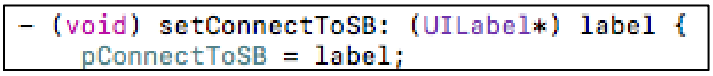
- 音樂播放

1. StoryBoard 拉連結到 GameViewController.h

2. 在 GameViewController.m synthesize

3. GameScene.m 宣告一個 UILabel 的 Pointer

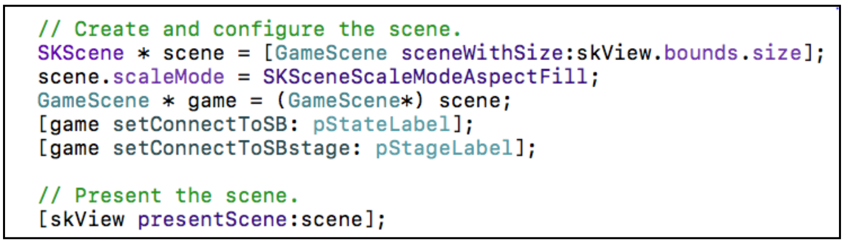
4. 寫個 Method 把 StoryBoard 上的 label pointer 傳到 GameScene



```
- (void) setConnectToSB: (UILabel*) label {
    pConnectToSB = label;
```

5. 到 GameView allocate SKView 的地方去呼叫 Method

6. 如果 SKView 不能呼叫 Method，把 SKView 轉成 GameScene

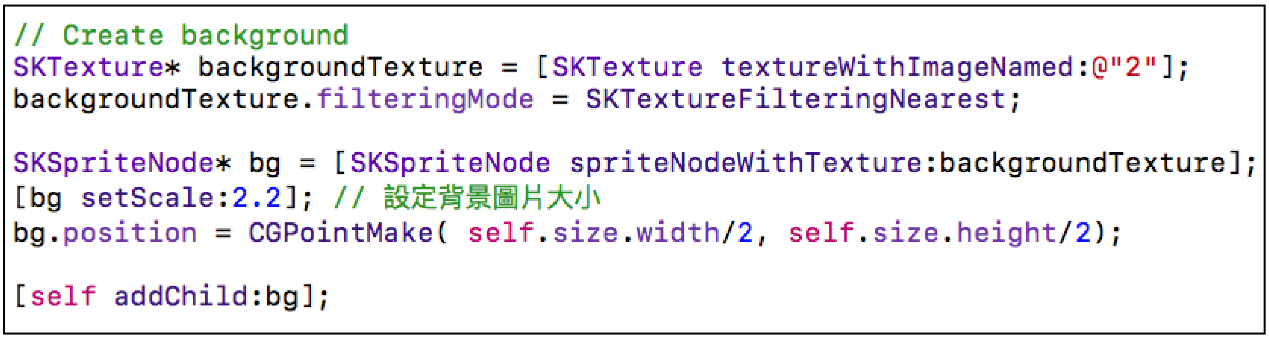


```
// Create and configure the scene.
SKScene * scene = [GameScene sceneWithSize:skView.bounds.size];
scene.scaleMode = SKSceneScaleModeAspectFill;
GameScene * game = (GameScene*) scene;
[game setConnectToSB: pStateLabel];
[game setConnectToSBstage: pStageLabel];

// Present the scene.
[skView presentScene:scene];
```

四、問題與解決方式

1. 背景把所有東西蓋住

解決:更改加入物件的順序



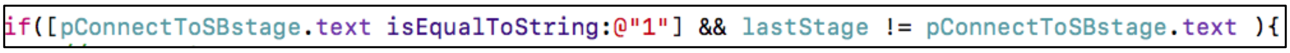
```
// Create background
SKTexture* backgroundTexture = [SKTexture textureWithImageNamed:@"2"];
backgroundTexture.filteringMode = SKTextureFilteringNearest;

SKSpriteNode* bg = [SKSpriteNode spriteNodeWithTexture:backgroundTexture];
[bg setScale:2.2]; // 設定背景圖片大小
bg.position = CGPointMake( self.size.width/2, self.size.height/2);

[self addChild:bg];
```

2. 切換說明頁，背景音樂播放不正常

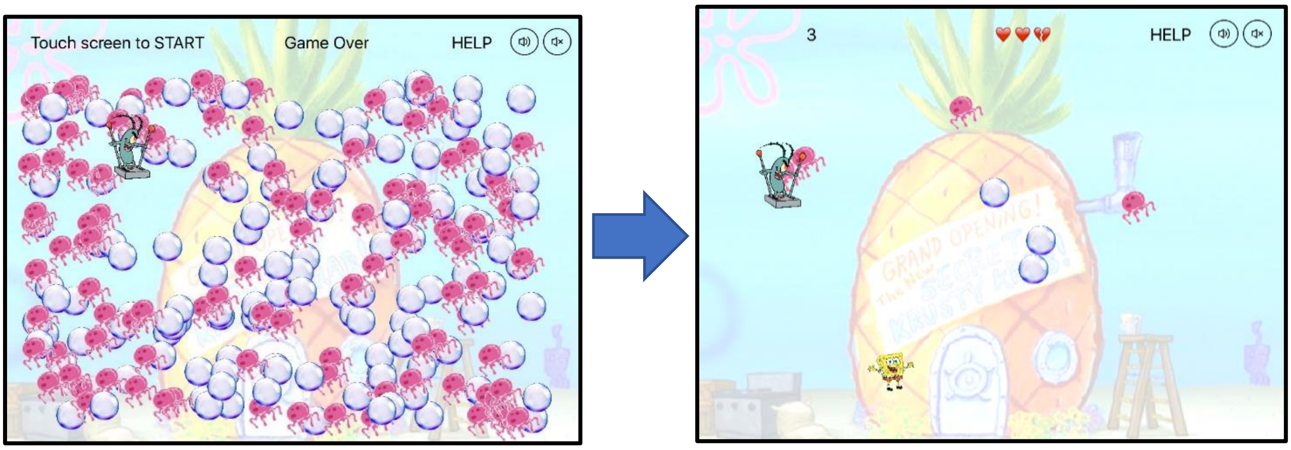
解決: 在兩頁之間傳遞參數，紀錄之前的設定

3. 東西無限出現

解決：更改 level 時，東西才更新（增加 laststage 變數來判斷）

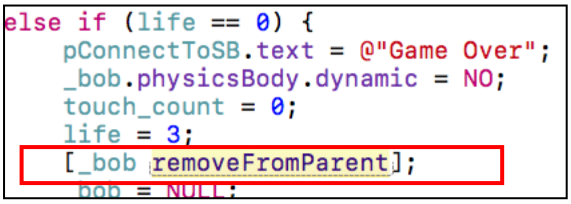


```
if([pConnectToSBstage.text isEqualToString:@"1"] && lastStage != pConnectToSBstage.text ){
```



4. 海綿寶寶死了，卻還在!

解決: 斷開與 Parent 的連結

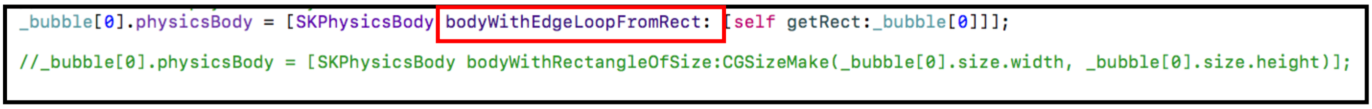


```
else if (life == 0) {
    pConnectToSB.text = @"Game Over";
    _bob.physicsBody.dynamic = NO;
    touch_count = 0;
    life = 3;
    [_bob removeFromParent];
    _bob = NULL;
```

5. 撞到東西，東西不會消失

解決: 更改設定物件邊界方式

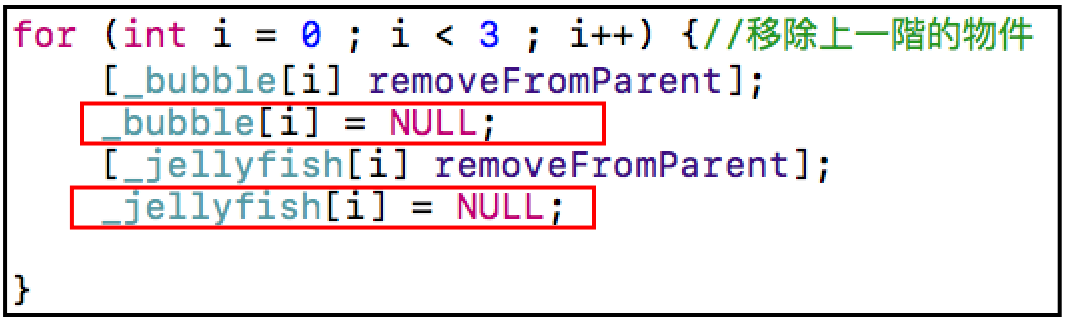
Circle/Rectangle → EdgeLoopFromRect



```
_bubble[0].physicsBody = [SKPhysicsBody bodyWithEdgeLoopFromRect: [self getRect:_bubble[0]]];

//_bubble[0].physicsBody = [SKPhysicsBody bodyWithRectangleOfSize:CGSizeMake(_bubble[0].size.width, _bubble[0].size.height)];
```

6. 撞到東西東西變透明但實際還在

解決:清除 Node (NULL)



```
for (int i = 0 ; i < 3 ; i++) {//移除上一階的物件
    [_bubble[i] removeFromParent];
    _bubble[i] = NULL;
    [_jellyfish[i] removeFromParent];
    _jellyfish[i] = NULL;

}
```

7. 切換說明頁，會出現許多 bug...

未解決