

*Object Oriented Programming*  
**Box2D Tutorials**  
**Tutorial #1 :建造一棟房子**

Shuo-Han Chen (陳碩漢),  
[shchen@csie.ntut.edu.tw](mailto:shchen@csie.ntut.edu.tw)

Hong-Yue Technology Research Building 1223  
F 09:10 - 12:00

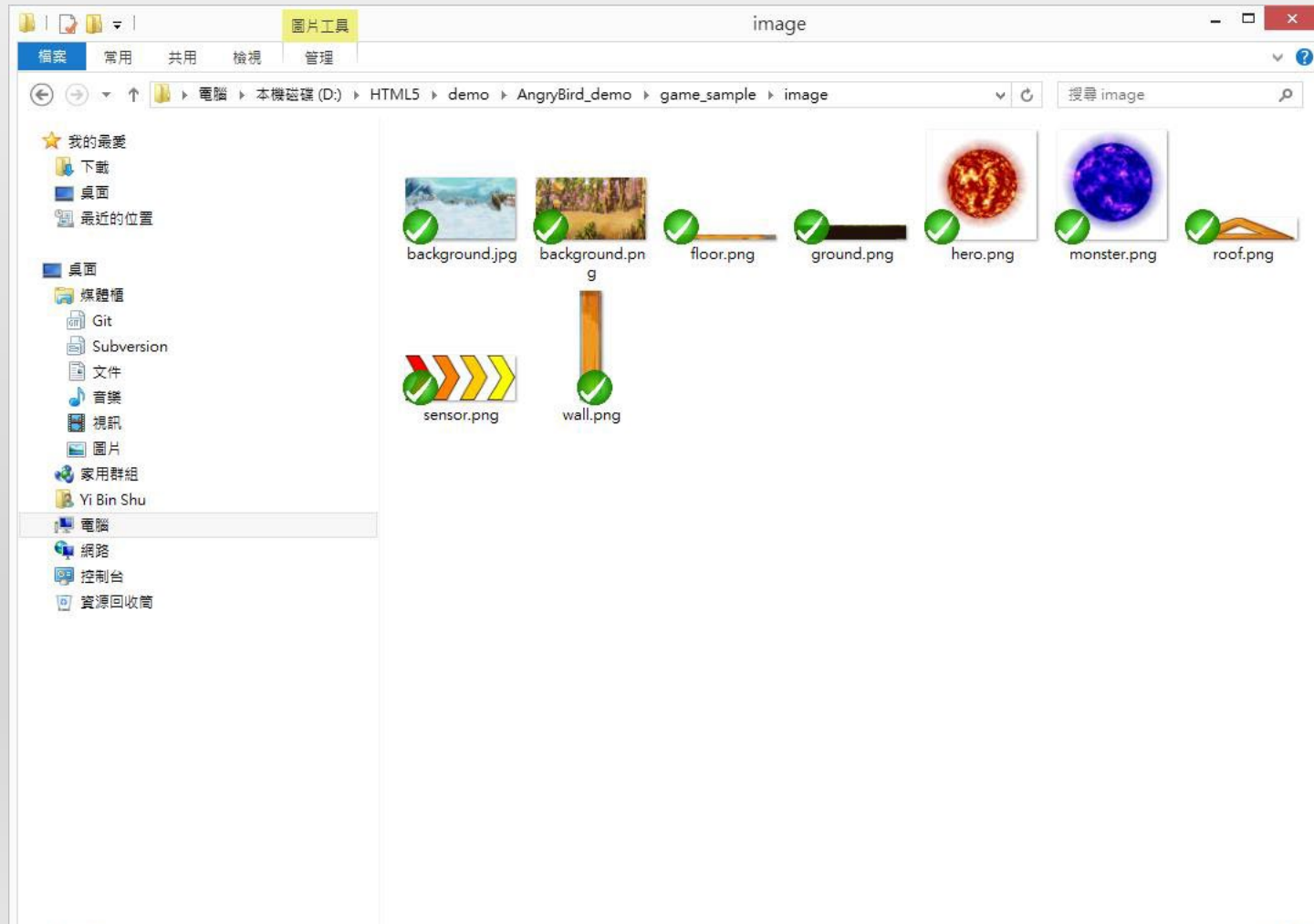
Part I.

# HTML5 Game Framework 使用說明

Tutorial 1 A : 在畫面中顯示圖片

1.打開 HTML5/demo/AngryBird\_demo

2.遊戲中會用到的圖片檔一律放置於 game\_sample/image資料夾中



### 3.背景圖片

打開game\_sample/js資料夾中的 MyBox2D.js, 此為遊戲主要模組，在 load function內需先載入遊戲中需要用到的圖片，載入遊戲背景圖並宣告座標

```
load : function () {  
    this.box2D = new Framework.Box2D(); //宣告Box2D物件  
    this.box2D.createWorld();           //產生Box2D世界  
  
    //背景圖  
    this.background = new Framework.Sprite(define.imagePath + 'background.jpg');  
    this.background.position = {  
        x : 800,  
        y : 450  
    };  
};
```

## 4.宣告物件：

專案目前需要三個物件

地板 (ground)、牆壁 (wall)、屋頂 (roof)

## 5.宣告地板物件：

在load function內宣告 ground物件，reateSquareBody為產生一矩形物件，參數分別為 (寬度，高度，靜態 or動態 )，並設置座標

```
//ground  
var ground = this.box2D.createSquareBody(1000, 1.0, this.box2D.bodyType_Static);  
ground.SetPosition(new this.box2D.b2Vec2(0, 24));  
|
```

## 6.宣告牆壁物件：

先在game\_sample/js資料夾內新增一 wall.js檔案，內容如下

```
var wall = function () {  
    this.arraySize = [];  
    this.component;  
  
    Object.defineProperty(this, 'position', {  
        get : function () {  
            this.component.position;  
        },  
        set : function (newValue) {  
            this.component.position = newValue;  
        },  
    });  
  
    Object.defineProperty(this, 'scale', {  
        get : function () {  
            this.component.scale;  
        },  
        set : function (newValue) {  
            this.component.scale = newValue;  
        },  
    });  
  
    Object.defineProperty(this, 'rotation', {  
        get : function () {
```

1

```
        Object.defineProperty(this, 'rotation', {  
            get : function () {  
                this.component.rotation;  
            },  
            set : function (newValue) {  
                this.component.rotation = newValue;  
            },  
        });
```

2

```
    });  
  
    Object.defineProperty(this, 'isSensor', {  
        get : function () {  
            this.component.isSensor;  
        },  
        set : function (newValue) {  
            this.component.isSensor = newValue;  
        },  
    });  
  
    this.init = function (sprite, box2D) {  
        this.pic = new Framework.Sprite(define.imagePath + sprite);  
        this.component = new Framework.squareComponent(this.pic,  
        box2D.bodyType_Dynamic, box2D);  
  
        this.component.fixtureDef.m_restitution = 0;  
        this.component.Body.m_userData = "wall";  
    };  
};
```

3

## 6.宣告牆壁物件：

```
”  
  
this.update = function () {  
    this.component.update();  
};  
  
this.draw = function () {  
    this.pic.draw();  
};  
}
```

4

其中defineProperty為物件的 get&set之用，主要在 init function中，component物件為 class中宣告 box2D物件的主體，在此宣告為一 矩形 動態物件



## 7.宣告屋頂物件：

先在game\_sample/js資料夾內新增一 roof.js檔案，內容如下

```
var roof = function () {  
    this.arraySize = [];  
    this.component;  
  
    Object.defineProperty(this, 'position', {
```

1

```
        get : function () {  
            this.component.position;  
        },  
        set : function (newValue) {  
            this.component.position = newValue;  
        },  
    });
```

```
    Object.defineProperty(this, 'scale', {  
        get : function () {  
            this.component.scale;  
        },  
        set : function (newValue) {  
            this.component.scale = newValue;  
        },  
    });
```

2

```
        Object.defineProperty(this, 'rotation', {  
            get : function () {  
                this.component.rotation;  
            },  
            set : function (newValue) {  
                this.component.rotation = newValue;  
            },  
        });
```

3

```
        Object.defineProperty(this, 'isSensor', {  
            get : function () {  
                this.component.isSensor;  
            },  
            set : function (newValue) {  
                this.component.isSensor = newValue;  
            },  
        });
```

```
        this.init = function (sprite, box2D) {  
            this.pic = new Framework.Sprite(define.imagePath + sprite);  
            this.component = new Framework.polygonComponent(this.pic,
```

## 7.宣告屋頂物件：

```
box2D.bodyType_Dynamic, box2D, this.arraySize);  
    this.component.fixtureDef.m_restitution = 0;  
    this.component.Body.m_userData = "roof";  
};  
  
this.update = function () {  
    this.component.update();  
};  
  
this.draw = function () {  
    this.pic.draw();  
};  
}
```

4

同wall class, 在此宣告為一多邊形動態物件，由 arraySize傳入的值決定多邊形的邊數

## 8.在 MyBox2D.js 加入宣告的物件：

在load function內先定義牆壁 & 屋頂座標

```
//walls position
this.wallsValue = [
    {x: 900, y: 500},
    {x: 1100, y: 500},
    {x: 1300, y: 500},
    {x: 1500, y: 500},
    {x: 1150, y: 200},
    {x: 1250, y: 200}
];
//roofs position
this.roofsValue = [
    {x: 1000, y: 300},
    {x: 1400, y: 300},
    {x: 1200, y: 150}
];
```

同在load function再宣告牆壁 & 屋頂的陣列並加上圖片、座標、縮放比例、旋轉角度等參數

```
this.walls = new Array();  
for(var i=0; i<this.wallsValue.length; i++){  
    this.walls[i] = new wall();  
    this.walls[i].init('wall.png', this.box2D);  
    this.walls[i].position = {
```

1

```
        x: this.wallsValue[i].x,  
        y: this.wallsValue[i].y  
    };  
    this.walls[i].scale = 1.0;  
    this.walls[i].rotation = 0;  
}
```

2

```
this.floor = new wall();  
this.floor.init('floor.png', this.box2D);  
this.floor.position = {  
    x: 1200,  
    y: 270  
};
```

3

```
this.roofs = new Array();  
for(var i=0; i<this.roofsValue.length; i++){  
    this.roofs[i] = new roof();  
    this.roofs[i].arraySize = [  
        new this.box2D.b2Vec2(-5, 1),  
        new this.box2D.b2Vec2(0, -1),  
        new this.box2D.b2Vec2(5, 1),  
    ];  
    this.roofs[i].init('roof.png', this.box2D);  
    this.roofs[i].position = {  
        x: this.roofsValue[i].x,  
        y: this.roofsValue[i].y  
    };  
}
```

## 9.加上遊戲的 initialize

將遊戲物件attach到遊戲畫面

```
initialize : function () {  
    //Scene Attach  
    this.rootScene.attach(this.background);  
    for(var i=0; i<this.walls.length; i++){  
        this.rootScene.attach(this.walls[i].pic);  
    }  
    this.rootScene.attach(this.floor.pic);  
    for(var i=0; i<this.roofs.length; i++){  
        this.rootScene.attach(this.roofs[i].pic);  
    }  
},
```

## 10.加上遊戲的 update&draw

將遊戲物件中需要update&draw加入 function

```
update : function () {  
    for(var i=0; i<this.walls.length; i++){  
        this.walls[i].update();  
    }  
  
    this.floor.update();  
  
    for(var i=0; i<this.roofs.length; i++){  
        this.roofs[i].update();  
    }  
    this.box2D.draw();  
},  
  
draw : function (parentCtx) {  
    this.box2D.draw();  
    this.rootScene.draw();  
},
```

## 11.載入宣告物件

在game\_sample/js/loadGame.js中需定義所有在遊戲中所需的物件檔案

```
//陣列和載入JS檔的順序相同，lookFor為在要載入的檔案中，  
//有用到的全域變數，importJS這個function，會在找到lookFor的變數後  
//才會繼續loading下一個檔案，如果沒有需要lookFor，則以空字串代表  
var listScript =  
[  
  { src: 'game_sample/js/define.js', lookFor: 'define' },  
  { src: 'game_sample/js/wall.js', lookFor: 'wall' },  
  { src: 'game_sample/js/roof.js', lookFor: 'roof' },  
  { src: 'game_sample/js/MyBox2D.js', lookFor: 'MyBox2D' },  
  { src: 'game_sample/js/mainGame.js'},  
]
```

## 12.執行結果

