

HTML5 Game Framework Box-2D 使用說明

Tutorial 3: 物理碰撞事件

1. 接續 Tutorial2

2. 註冊 Box2D ContactListener

在 game_sample/js/MyBox2D.js 的 load function 內增加 setContactListener

```
load : function () {  
    this.box2D = new Framework.Box2D ();  
    this.box2D.createWorld ();  
    this.box2D.setContactListener ();  
}
```

3. 宣告 monster 物件：

在 game_sample/js 資料夾內新增一 monster.js 內容如下

```
var monster = function () {  
    this.arraySize = [];  
    this.component;  
    this.mbox2D;  
    this.isDead = false;  
  
    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 () {
    this.component.rotation;
  },

  set : function (newValue) {
    this.component.rotation = newValue;
  },
});

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

  set : function (newValue) {
    this.component.isSensor = newValue;
  },
});

var mMonster = this;
this.contactCallback = function(bodyB, force){
  if(Math.abs(force) > 5){
    mMonster.dead();
  }
}

this.init = function (sprite, box2D) {
  this.mbox2D = 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.registerContact(this.contactCallback);
  this.component.Body.m_userData = "monster";
};

```

```
    this.update = function () {
        this.component.update();
        if(this.isDead){
            this.mbox2D.world.DestroyBody(this.component.Body);
        }
    };

    this.draw = function () {
        this.pic.draw();
    };

    this.dead = function(){
        this.mbox2D.world.DestroyBody(this.component.Body);
        this.isDead = true;
    }
}
```

其中 **this.contactCallBack** 為碰撞事件觸發時呼叫的 function, 當碰撞力道大於一定時觸發 **this.dead**

4. 在 **MyBox2D.js** 加入宣告的物件：

在 load function 內先宣告 monster 並定義其座標

```
this.monstersValue = [
    {x: 1000, y: 550},
    {x: 700, y: 550}
];
this.monsters = new Array();
for(var i=0; i<this.monstersValue.length; i++){
    this.monsters[i] = new monster();
    this.monsters[i].init('monster.png', this.box2D);
    this.monsters[i].position = {
        x: this.monstersValue[i].x,
        y: this.monstersValue[i].y
    };
}
```

5. 加上遊戲的 initialize

將遊戲物件 attach 到遊戲畫面

```

for(var i=0; i<this.monsters.length; i++){
    this.rootScene.attach(this.monsters[i].pic);
}

```

6. 加上遊戲的 update

將遊戲物件中需要 update 加入 function

```

update : function () {
    this.angryBird.update();
    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();
    }

    for(var i=0; i<this.monsters.length; i++){
        this.monsters[i].update();
    }

    this.box2D.draw();
},

```

7. 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/angryBird.js', lookFor: 'angryBird' },
    { src: 'game_sample/js/monster.js', lookFor: 'monster' },
    { src: 'game_sample/js/MyBox2D.js', lookFor: 'MyBox2D' },
    { src: 'game_sample/js/mainGame.js' },
]

```

8. 執行結果

