

HTML5 Game Framework Box-2D 使用說明

Tutorial 2: 基本物理特性

1. 接續 Tutorial1

2. 宣告 angryBird 物件：

在 game_sample/js 資料結內新增一 angryBird.js 內容如下

```
var angryBird = function () {
    this.arraySize = [];
    this.component;
    this.mbox2D;

    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;
    },
});

this.init = function (sprite, box2D) {
    this.mbox2D = box2D;
    this.pic = new Framework.Sprite(define.imagePath + sprite);
    this.component = new Framework.circleComponent(this.pic,
box2D.bodyType_Dynamic, box2D);
    this.component.fixtureDef.m_restitution = 0;
    this.component.Body.m_userData = "angryBird";
};

this.update = function () {
    this.component.update();
};

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

this.shoot = function (angle) {
    var degrees = angle-90;
    var power = 4000;
    this.component.Body.ApplyForce(new
this.mbox2D.b2Vec2(Math.cos(degrees * (Math.PI / 180)) * power,
                    Math.sin(degrees * (Math.PI /
180)) * power),

```

```
        this.component.Body.GetWorldCenter());  
        };  
    }
```

其中 `this.shoot function` 表示給予物件固定 4000 的力，並向參數 `angle` 的角度施力。

3. 在 `MyBox2D.js` 加入宣告的物件：

在 `load function` 內先宣告 `angryBird` 並定義其座標

```
//angryBird  
this.angryBird = new angryBird();  
this.angryBird.init('hero.png', this.box2D);  
this.angryBird.position = {  
    x: 180,  
    y: 100  
};
```

4. 加上遊戲的 `initialize`

將遊戲物件 `attach` 到遊戲畫面

```
this.rootScene.attach(this.angryBird.pic);
```

5. 加上遊戲的 `update`

將遊戲物件中需要 `update` 加入 `function`

```

update: function() {
    //angryBird
    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();
    }

    this.box2D.draw();
},

```

6. 增加 Click 事件

在 game_sample/js/MyBox2D.js 增加 Click function, 內容如下

```

click: function(e){
    angle = Math.atan2(e.y-
    this.angryBird.component.position.y, e.x-
    this.angryBird.component.position.x);
    angle = angle * 180 / Math.PI+90;
    this.angryBird.shoot(angle);
},

```

在滑鼠按下的事件執行 angryBird 的 shoot function 並計算滑鼠按下座標和 angryBird 座標角度

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/MyBox2D.js', lookFor: 'MyBox2D' },  
  { src: 'game_sample/js/mainGame.js' },  
]
```

8. 執行結果

