

## Pd =HTML5 Game Framework 使用說明

### Tutorial 3: 將圖片與動作製成物件

1. 請先完成 Tutorial2
2. 在 `OOPLab_Sample\game_sample\js` 下新增一個 `js` 檔，命名為 `Practice`，並且實作 `load`, `initialize`, `update` 跟 `draw`

```
var Practice = function() {
  this.load = function() {
    this.pic = new Framework.Sprite(define.imagePath + '169.bmp');
    this.pic.position = {
      x: 100,
      y: 100
    }
    this.pic.rotation = 0;
    this.position = {
      x: 100,
      y: 100
    }
    this.rotation = 0;
  };

  this.initialize = function() {

  };

  this.update = function() {
    this.position = {
      x: this.position.x + 1,
      y: this.position.y
    }
    this.rotation += 1;
    this.pic.position = this.position
    this.pic.rotation = this.rotation
  };

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

3. 在 `loadGame.js` 中加入 `Practice.js`

```

//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/myMenu.js', lookFor: 'MyMenu' },
  { src: 'game_sample/js/character.js', lookFor: 'Character' },
  { src: 'game_sample/js/Practice.js', lookFor: 'Practice' },
  { src: 'game_sample/js/myGameLevel1.js', lookFor: 'MyGame' },
  { src: 'game_sample/js/mainGame.js' }
]

```

#### 4. 在 myGameLevel1.js load function 宣告 Practice 物件

```

this.practice = new Practice();
this.practice.load();
this.rootScene.attach(this.practice.pic);

```

update function 增加 practice.update()

```

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

```

#### 5. 執行結果

