

# 實務專題計畫成果摘要報告

## 海石榴華SVG反應音樂動畫

專題編號:108-CSIE-S025

執行期限:107 年第 1 學期至 108 年第 1 學期

指導教授: 謝東儒 教授

專題參與人員: 105590048 羅允宣

### 一、摘要

此系統透過將音樂與svg圖片結合起來達到音樂可視化。操作方法簡單易上手，首先繪製圖片，保存後再添加音樂。當速度變快時，隨著音樂會顯示較快的可視化屏幕，反之可視化屏幕會變慢。

### 二、緣由與目的

我們知道當 2D 圖片移動會形成動畫，如果通過將這樣的動畫與音樂相結合，也就是根據音樂節奏移動圖片，則將會更加生動。

### 三、工具與使用技術方法

#### (一) SMIL

同步多媒體集成語言是W3C為採用XML描述多媒體而提出的建議標準。它定義了時間標籤、佈局標籤、動畫、視覺漸變和媒體嵌入等。

#### (二) SVG

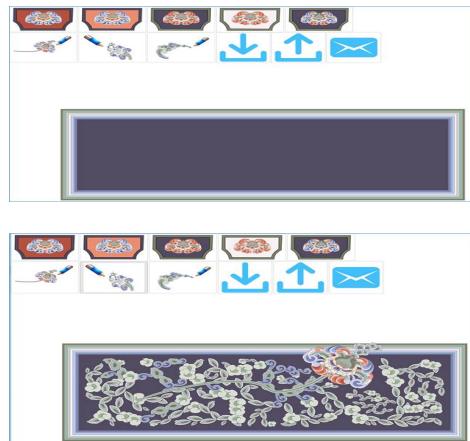
SVG 的動畫元素是由 SMIL，同步多媒體集成語言所規範開發，SMIL 規劃具有許多的參數與設定值，也因為有了這些設定，我們便可以輕鬆的控制 SVG圖形這篇 將介紹 SVG基本的動畫，進階動畫控制的部分會在下一篇文章進行分享.

### 四、使用技術

網頁使用HTML、JavaScript, svg.使用jsfft 到可視化聲音.

### 五、架構流程

(一) 首先，在網站上畫一幅圖畫並保存。



## (二) 打開保存的圖片，使用 svg

animation效果使圖片移動。

### animateMotion

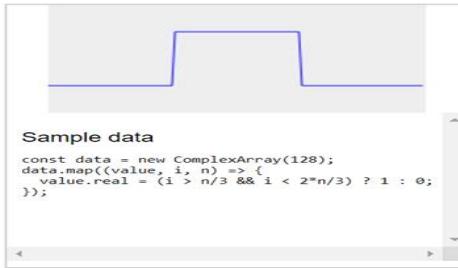
這個屬性是讓我們的圖形，可以跟隨路徑移動，甚至可以設定跟隨路徑自動旋轉，如果對於路徑 path 還不熟悉的，可以參考 [SVG 研究之段 \(4\) - Path 基礎篇與 SVG 研究之路 \(5\) - Path 路徑篇](#)，比較需要注意的是，當我們設定了 animateMotion 屬性時，則圖形位置以 path 的起點 M 開始計算，也就是會把 M 當作圓形的 (0,0) 的座標。因此下面的範例將 x 設為 0，y 設為 -10。

```
<rect width="20" height="10" x="0" y="-10" fill="#09c">
</animateMotion>
<path fill="none" stroke="#000" d="M7.4,15.3c17.20,4.48,8.38,91.6,27.8c79.5-18.9,107.4,48.2c-10.4,34.3-20.4,38.9z" style="stroke-dasharray: 5 5;">
</path>
```



```
11  var animateCount = 0;
12  var animators = [
13    {
14      "http://www.w3.org/2009/svg#": "animateTransform",
15      "x": "0px", "y": "0px", "x2": "0px", "y2": "0px", "dur": "5s", "path": "M7.4,15.3c17.20,4.48,8.38,91.6,27.8c79.5-18.9,107.4,48.2c-10.4,34.3-20.4,38.9z", "type": "rotate", "values": "0deg, 360deg", "keyTimes": "0,1", "fill": "none", "stroke": "#000", "strokeDasharray": "5 5", "strokeWidth": "1px", "strokeDashoffset": "0", "startOffset": "0", "addition": "sum"
16    },
17    {
18      "http://www.w3.org/2009/svg#": "animateMotion",
19      "x": "0px", "y": "0px", "x2": "0px", "y2": "0px", "dur": "5s", "path": "M7.4,15.3c17.20,4.48,8.38,91.6,27.8c79.5-18.9,107.4,48.2c-10.4,34.3-20.4,38.9z", "type": "translate", "values": "0, 0", "keyTimes": "0,1", "fill": "none", "stroke": "#000", "strokeDasharray": "5 5", "strokeWidth": "1px", "strokeDashoffset": "0", "startOffset": "0", "addition": "sum"
20    }
21  ];
22  animators[0].start();
23  animators[1].start();
24  animators[2].start();
25  animators[3].start();
26  animators[4].start();
27  animators[5].start();
28  animators[6].start();
29  animators[7].start();
30  animators[8].start();
31  animators[9].start();
32  animators[10].start();
33  animators[11].start();
34  animators[12].start();
35  animators[13].start();
36  animators[14].start();
37  animators[15].start();
38  animators[16].start();
39  animators[17].start();
40  animators[18].start();
41  animators[19].start();
42  animators[20].start();
43  animators[21].start();
44  animators[22].start();
45  animators[23].start();
46  animators[24].start();
47  animators[25].start();
48  animators[26].start();
49  animators[27].start();
50  animators[28].start();
51  animators[29].start();
52  animators[30].start();
53  animators[31].start();
54  animators[32].start();
55  animators[33].start();
56  animators[34].start();
57  animators[35].start();
58  animators[36].start();
59  animators[37].start();
60  animators[38].start();
61  animators[39].start();
62  animators[40].start();
63  animators[41].start();
64  animators[42].start();
65  animators[43].start();
66  animators[44].start();
67  animators[45].start();
68  animators[46].start();
69  animators[47].start();
70  animators[48].start();
71  animators[49].start();
72  animators[50].start();
73  animators[51].start();
74  animators[52].start();
75  animators[53].start();
76  animators[54].start();
77  animators[55].start();
78  animators[56].start();
79  animators[57].start();
80  animators[58].start();
81  animators[59].start();
82  animators[60].start();
83  animators[61].start();
84  animators[62].start();
85  animators[63].start();
86  animators[64].start();
87  animators[65].start();
88  animators[66].start();
89  animators[67].start();
90  animators[68].start();
91  animators[69].start();
92  animators[70].start();
93  animators[71].start();
94  animators[72].start();
95  animators[73].start();
96  animators[74].start();
97  animators[75].start();
98  animators[76].start();
99  animators[77].start();
100 animators[78].start();
101 animators[79].start();
102 animators[80].start();
103 animators[81].start();
104 animators[82].start();
105 animators[83].start();
106 animators[84].start();
107 animators[85].start();
108 animators[86].start();
109 animators[87].start();
110 animators[88].start();
111 animators[89].start();
112 animators[90].start();
113 animators[91].start();
114 animators[92].start();
115 animators[93].start();
116 animators[94].start();
117 animators[95].start();
118 animators[96].start();
119 animators[97].start();
120 animators[98].start();
121 animators[99].start();
122 animators[100].start();
123 animators[101].start();
124 animators[102].start();
125 animators[103].start();
126 animators[104].start();
127 animators[105].start();
128 animators[106].start();
129 animators[107].start();
130 animators[108].start();
131 animators[109].start();
132 animators[110].start();
133 animators[111].start();
134 animators[112].start();
135 animators[113].start();
136 animators[114].start();
137 animators[115].start();
138 animators[116].start();
139 animators[117].start();
140 animators[118].start();
141 animators[119].start();
142 animators[120].start();
143 animators[121].start();
144 animators[122].start();
145 animators[123].start();
146 animators[124].start();
147 animators[125].start();
148 animators[126].start();
149 animators[127].start();
150 animators[128].start();
151 animators[129].start();
152 animators[130].start();
```

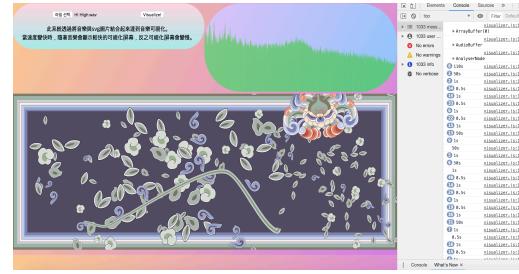
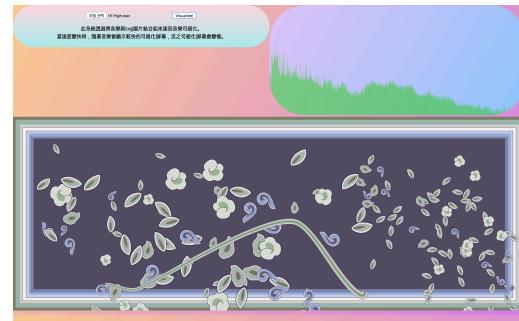
## (三) 使用 jsfft, 根據音樂的節奏可視化保存的圖像。



```
111 function Visualize(){
112   for (var i = 0; i < animateCount; i++) {
113     document.getElementById("animate" + i).beginElement();
114   }
115   if (audioSourceNode) {
116     audioSourceNode.stop();
117     for (var i = 0; i < animateCount; i++) {
118       document.getElementById("animate" + i).endElement();
119     }
120   }
121   audioContext = new AudioContext();
122   var input = document.getElementById("audio-file");
123   var file = input.files[0];
124   var fr = new FileReader();
125   fr.readAsArrayBuffer(file);
126   fr.onload = function() {
127     var buffer = fr.result;
128     console.log(buffer);
129     audioContext.decodeAudioData(buffer).then(function(decodedData) {
130       console.log(decodedData);
131       audioSourceNode = audioContext.createBufferSourceNode(audioContext);
132       audioSourceNode.buffer = decodedData;
133       audioSourceNode.start();
134       AnalyseAudio(audioSourceNode);
135     });
136   }
137 }
138
139 function AnalyseAudio(audioSourceNode) {
140   bufferSize = 4096;
141   freqInCents = bufferSize / 2;
142   analyser = new AnalyserNode(audioContext);
143   analyser.fftSize = bufferSize;
144   analyser.smoothingTimeConstant = 0.05;
145
146   audioSourceNode.connect(analyser);
147   analyser.connect(audioContext.destination);
148
149   console.log(analyser);
150   dataArray = new Float32Array(bufferSize / 2);
151   window.requestAnimationFrame(draw);
152 }
```

## (三) 音樂與svg圖片結合起來達到音樂可視化。,

隨著音樂會顯示較快的可視化屏幕，反之可視化屏幕會變慢。



## 六、結論

此系統旨在讓使用者的圖片更加生動，並期望能成為一個對於在視覺、聽覺、感官及情感的發展上能有所幫助的系統

## 七、參考文獻

[1]

研究生：劉士豪

研究生(外文)：Shih-Hao Liu

論文名稱：營造法式海石榴華圖樣生成

論文名稱(外文)：Chinese Sea Pomegranate Granatum Floral Pattern Synthesis

指導教授：謝東儒謝東儒引用關係

指導教授(外文)：Tung-Ju Hsieh

校院名稱：國立臺北科技大學

系所名稱：資訊工程系研究所

學門：工程學門

學類：電資工程學類

語文別：中文