

第4章 カスタマイズ(2)

Web応用

第12回 Webページの図形の描画3～アニメーション

第4章

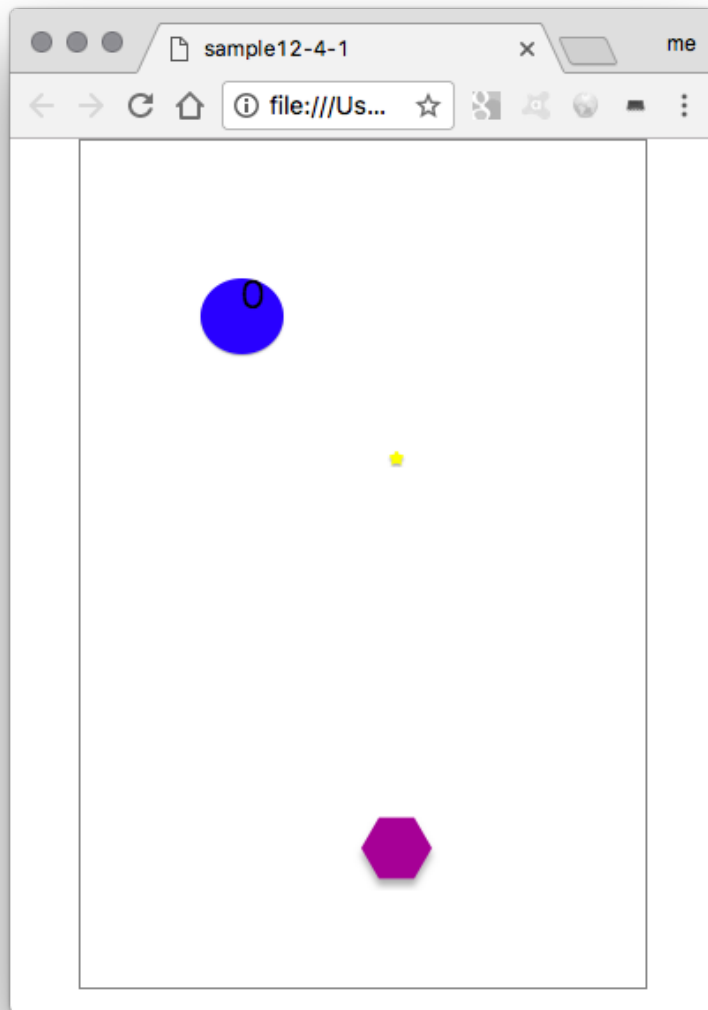
カスタマイズ(2)

第4章 学習目標

図形を画像に変更する方法について理解できる。

1. 画像を描画

canvasでは画像を表示できることを学びました。それをこのアニメーションに応用してみましょう。そうするとより豊かな表現ができるようになります。



1. ファイルの準備

次のファイルを準備しましょう(sample12-4-1.html)

■ サンプル

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>12-4-1</title>
6   <style>
7     * {margin:0; padding:0;}
8     p{text-align:center;}
9     #canvas1{border:1px solid gray;}
10  </style>
11 </head>
12 <body>
13   <p><canvas id="canvas1" width="320" height="480"></canvas></p>
14
15   <script>
```

```
16 var canvas1 = document.getElementById("canvas1");
17 var ctx1 = canvas1.getContext("2d");
18
19 //変数の設定と初期化
20 var target = {x:250, y:100, dir:1};
21 var player = {x:250, y:400};
22 var ball = {x:-10, y:-10};
23 var point = 0;
24
25 //一定時間毎起動させる
26 setInterval(game1,50);
27
28 //game1関数
29 function game1(){
30     ctx1.clearRect(0,0,320,480);
31
32     //ターゲット
33     ctx1.fillStyle = "black";
34     ctx1.fillRect(target.x-25,target.y-5,50,10);
35
36     //得点
37     ctx1.fillStyle = "black";
38     ctx1.font = "20px sans-serif ";
39     ctx1.fillText(point,target.x,target.y-5);
40
41     //プレーヤ
42     ctx1.fillStyle = "blue";
43     ctx1.fillRect(player.x-25,player.y-5,50,10);
44
45     //描画(ボール)
46     ctx1.fillStyle = "red";
47     ctx1.beginPath();
48     ctx1.arc(ball.x,ball.y,5,0,2*Math.PI,false);
49     ctx1.closePath();
50     ctx1.fill();
51
52     //移動計算
53     target.x += target.dir;
54     ball.y -= 5;
55
56     if(target.x < 25 || 295 < target.x){target.dir *= -1;}
57
58     //得点
59     if ((ball.y === 100) && (Math.abs(target.x - ball.x)<25)){
60         point++;
61     }
62 }
63
64 //シュート
65 canvas1.addEventListener("mousedown",function(){
66     ball.x = player.x;
67     ball.y = player.y;
68 });
```

```

69
70 //playerコントロール
71 canvas1.addEventListener("mousemove",function(e){
72     var rect = e.target.getBoundingClientRect();
73     player.x = e.clientX-rect.left;
74 });
75 </script>
76 </body>
77 </html>

```

2. 画像の準備

画像のファイルを準備します。以下、ダミーを提供しますが自作してもかまいません。

- target.png (縦横50px)
- player.png (縦横50px)
- ball.png (縦横10px)

3. 初期化に追加

初期化のところに以下を追記して画像を読み込みます。

■ サンプル

```

19 //変数の設定と初期化
20 var target = {x:250, y:100, dir:1};
21 var player = {x:250, y:400};
22 var ball = {x:-10, y:-10};
23 var point = 0;
24
25 var targetImage=new Image();
26 var playerImage=new Image();
27 var ballImage=new Image();
28
29 targetImage.src="target.png";
30 playerImage.src="player.png";
31 ballImage.src="ball.png";

```

4. target、player、ballを変更


それぞれ、矩形、円弧から「画像の描画」に書き換えます。

■ サンプル

```

37 //ターゲット
38 ctx1.drawImage(targetImage, target.x-25, target.y-25, 50, 50);
39
40 //プレーヤ
41 ctx1.drawImage(playerImage, player.x-25, player.y-25, 50, 50);
42
43 //ボール
44 ctx1.drawImage(ballImage, ball.x-5, ball.y-5, 10, 10);

```

前回学んだ.onload=function(){}

がないのを不思議に思う人もいると思います。
アニメーションの場合は、繰り返し描画するので、必要ありません。

2. 効果音を追加

得点が入った時に、効果音ができるようにしましょう。

1. 音声の準備

音声のファイルを準備します。以下、音声ファイルを提供しますが自作してもかまいません。

- audio1.mp3

2. 音声ファイルの読み込み

「変数の設定と初期化」部分に次のコードを追記して音声を読み込みます。

■ サンプル

```

32 var audio1=new Audio();
33 audio1.src="audio1.mp3";

```

- 前はaudio要素を設置して要素の取得を行っていましたが、このように、音声のコンストラクタ「Audio()」を使ってオブジェクト「audio1」を生成することもできます。音声ファイルの指定は「.src」を用いて指定します。

3. 再生

得点時に音を鳴らします。

■ サンプル

```

56 //得点
57 if ((ball.y === 100) && (Math.abs(target.x - ball.x)<25)){
58     point++;
59     audio1.play();
60 }

```

3. 12回科目レポート・ディベートについて

私たちは「Canvas API」を使ってゲームアプリケーションを制作しました。そこで、

- 「科目レポート」では、その制作したものに一工夫（オリジナル化）して提出してください。
- 「科目ディベート」では、制作の過程で苦労したこと、工夫したこと、検索して調べたことなどを投稿してください。

アプリケーションの要件や判断基準などは、第12回の「科目レポート」「科目ディベート」を参照してください。

期限をすぎると遅刻扱いになり、減点されますので余裕をもって提出してください。

私は「皆さん自身がクリエイトしたオリジナルなアプリケーション」を期待しています。

練習問題1

問題

【クイズ】 択一選択（即解答表示）

canvasのアニメーションで画像を表示するとき、注意しなければならないのはどれか。

- ☐ 画像を読み込んでから実行する「img.onload=function(){〜}」は不要。
- ☐ 画像を読み込んでから実行する「img.onload=function(){〜}」を忘れると表示できない。

練習問題1の解説

正解は

画像を読み込んでから実行する「img.onload=function(){〜}」は不要。

です。

アニメーションではない画像を表示する際は、画像の読み込みに時間がかかるので必要ですが、アニメーションの場合は必要ではありません。

第4章 まとめ

図形を画像に変更して表示する方法について理解できた。

第4章 終わり

Web応用

第12回 Webページの図形の描画3～アニメーション

第4章

カスタマイズ(2)

終わり