

第3章 カスタマイズ(1)

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3章

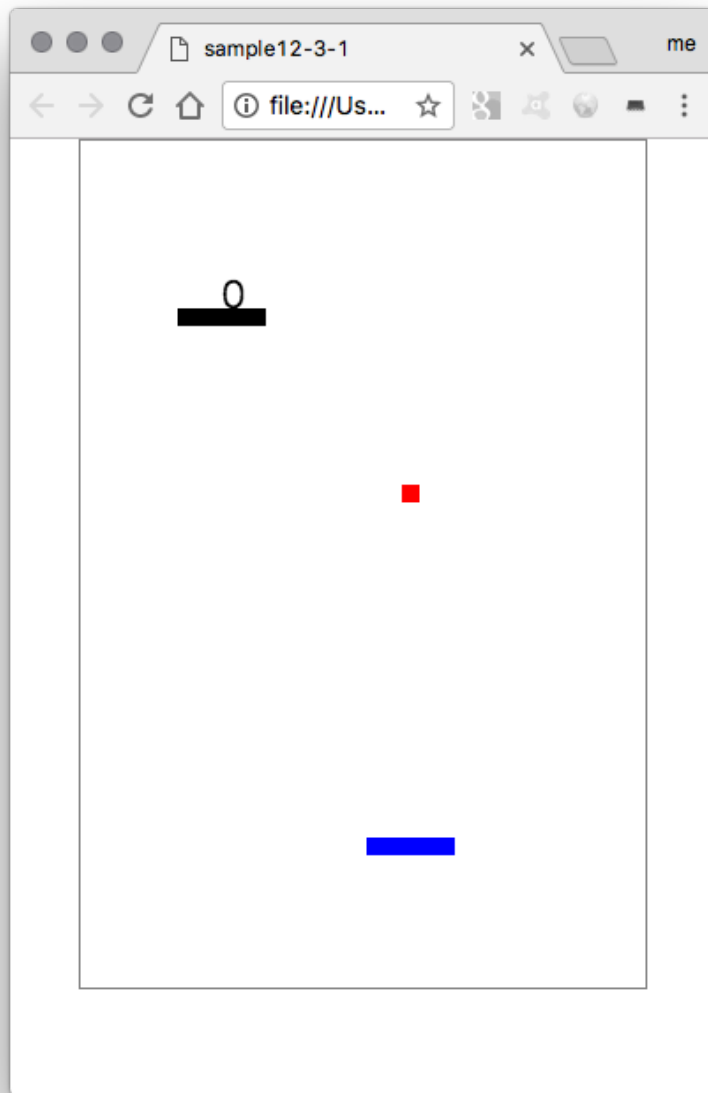
カスタマイズ(1)

第3章 学習目標

背景、図形のカスタマイズについて理解できる。

1. ゲームアプリ化

Canvas APIを使って、スマートフォンやPCで楽しめるようなゲームアプリケーションを制作しましょう。



1. ファイルの準備

次のファイルを準備しましょう(sample12-3-1.html)

■ サンプル

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>12-3-1</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     //①変数の設定と初期化
20
21     //②一定時間毎にプログラムgame1を起動させる
22
23     //③game1関数
24     function game1(){
25         //画面をクリア
26
27         //描画(ターゲット)
28
29         //描画(得点)
30
31         //描画(プレイヤー)
32
33         //描画(ボール)
34
35         //移動計算
36
37         //壁衝突計算
38
39         //得点計算
40
41     }
42
43     //④シュート関数
44     canvas1.addEventListener("mousedown",function(){
45
46     });
47
48     //⑤プレイヤーコントロール
49     canvas1.addEventListener("mousemove",function(e){
50
51     });
52
53 </script>
54 </body>
55 </html>

```

解説

canvasでアニメーションを行いながら、addEventListener()でマウスのボタンを押したとき(mousedown)とマウスを移動させたとき(mousemove) にそれぞれ、playerの横方向の位置やballの位置をコントロールしています。

2. 初期化

変数を用意し、初期値を代入します。

■ サンプル

```
19 //①変数の設定と初期化
20 var target = {x:250, y:100, dir:1}; // ターゲットのxy位置と移動方向
21 var player = {x:250, y:400}; // プレーヤーのxy位置
22 var ball = {x:-10, y:-10}; // ボールのxy位置
23 var point = 0; // 得点
```

- 上空を飛んでいるものを「target」、私達を「player」、playerからtargetへ飛んで行くものを「ball」、得点を「point」とします。
- それぞれに値を代入して初期化します。

3. setInterval()

一定時間ごとに描画するためのsetInterval()を記述します。

■ サンプル

```
25 //②一定時間毎にプログラムgame1を起動させる
26 setInterval(game1,50);
```

4. 関数の定義

画面を消去し、描画し、次の位置などを計算する関数「game1()」を定義します。

■ サンプル

```

27 //game1関数
28 function game1(){
29     //画面をクリア
30     ctx1.clearRect(0,0,320,480);
31
32     //描画(ターゲット)
33     ctx1.fillStyle = "black";
34     ctx1.fillRect(target.x-25,target.y-5,50,10);
35
36     //描画(得点)
37     ctx1.fillStyle = "black";
38     ctx1.font = "20px sans-serif";
39     ctx1.fillText(point,target.x,target.y-5);
40
41     //描画(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.fillRect(ball.x-5,ball.y-5,10,10);
48
49     //移動計算
50     target.x += target.dir;
51     ball.y -= 5;
52
53     //壁衝突計算
54     if(target.x < 25 || 295 < target.x){target.dir *= -1;}
55
56     //得点計算
57     if ((ball.y === 100) && (Math.abs(target.x - ball.x)<25)){
58         point++;
59     }
60 }

```

- Math.abs() : 絶対値を求めます。これによりtargetとballのx方向の距離を正の数で求めることができます
- && : 「且つ」という意味です。ballのy座標が100に到達した時に、targetとballのx軸方向の距離が25未満だったら、ポイントを1点追加することができます。
- point++ : 実行された時にpointの値を1だけ増加させます。

5. イベント処理(シュート)

マウスボタンを押した時にballがplayerの位置にくるようにします。

■ サンプル

```
62 //④シュート関数
63 canvas1.addEventListener("mousedown",function(){
64     ball.x = player.x;
65     ball.y = player.y;
66 });
```

6. イベント処理(player)

マウスがcanvasの上で移動した時に、マウスのx座標の値でplayerを移動させます。

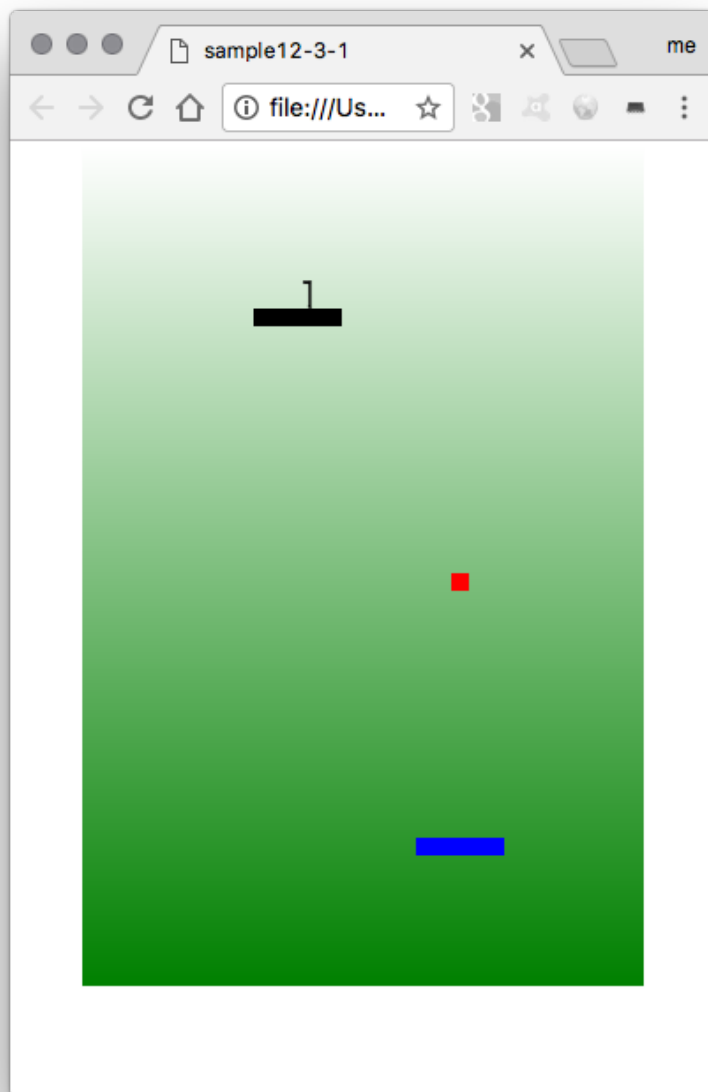
■ サンプル

```
68 //⑤playerコントロール
69 canvas1.addEventListener("mousemove",function(e){
70     var rect = e.target.getBoundingClientRect();
71     player.x = e.clientX-rect.left;
72 });
```

2. 背景を変更

canvasの背景をCSSで設定してみましょう。

- 学習教材用に、背景画像を提供します。(bg1.png)



背景の設定

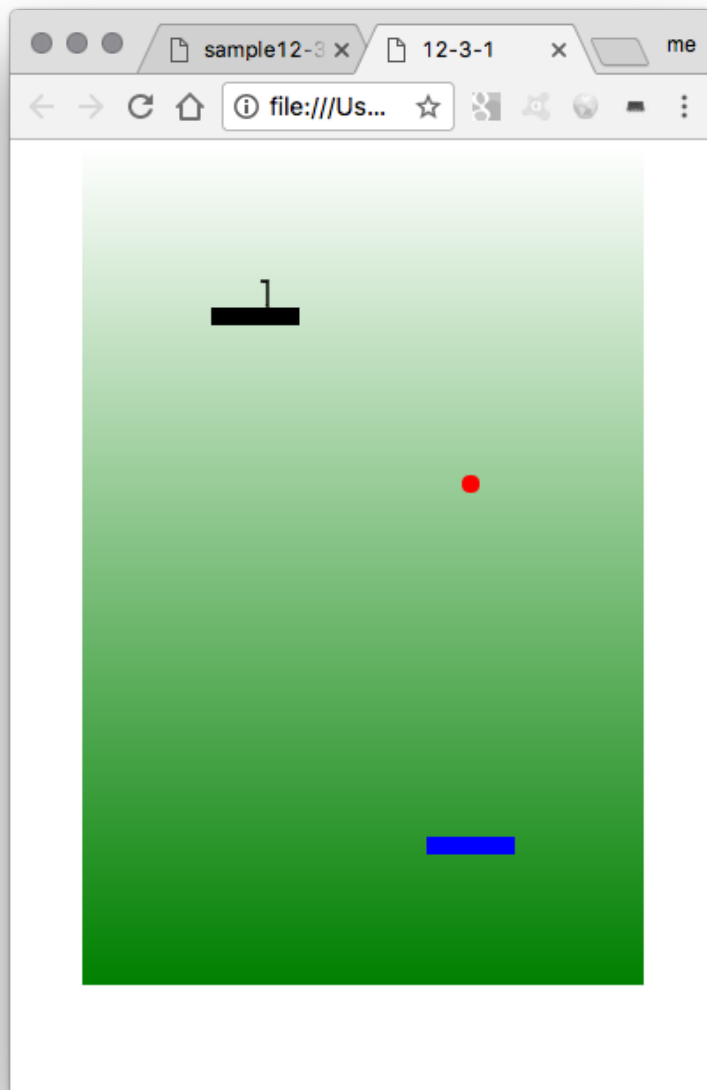
CSSで背景を設定します。style要素内を次のように変更しましょう。

■ サンプル

```
6 <style>
7   * {margin:0; padding:0;}
8   p{text-align:center;}
9   #canvas1{
10      background-image:url(bg1.png);
11      background-size:100% 100%;
12   }
13 </style>
```

3. 図形を変更

私たちは、様々な図形の描き方を学びました。ballを矩形から円に変形しましょう。



1. 図形の変更

以下のようにコード変更します。

■ サンプル

```
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();
```


2. 発展

player、target、ballを他の図形に変更してみましょう。

canvasには、今まで学んだ以外にも、曲線、グラデーション、変形、合成なども可能です。興味のある人は次のページで学びを深めてください。

https://developer.mozilla.org/ja/docs/Web/Guide/HTML/Canvas_tutorial

練習問題1

問題

【クイズ】 択一選択（即解答表示）

.addEventListener()で用いた「mousedown」はどのようなイベントですか。

- ☐ マウスを移動させたとき
- ☐ マウスのボタンを押したとき
- ☐ マウスがその要素に乗ったとき

練習問題1の解説

正解は

マウスのボタンを押したとき

です。これにより、マウスのボタンでもアプリケーションを操作ができるようになりましたね。

第3章 まとめ

ゲームアプリケーションの制作を通じて、背景、図形のカスタマイズについて理解できた。

第3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3章

カスタマイズ(1)

終わり