

「ブロック崩し」カスタマイズ例

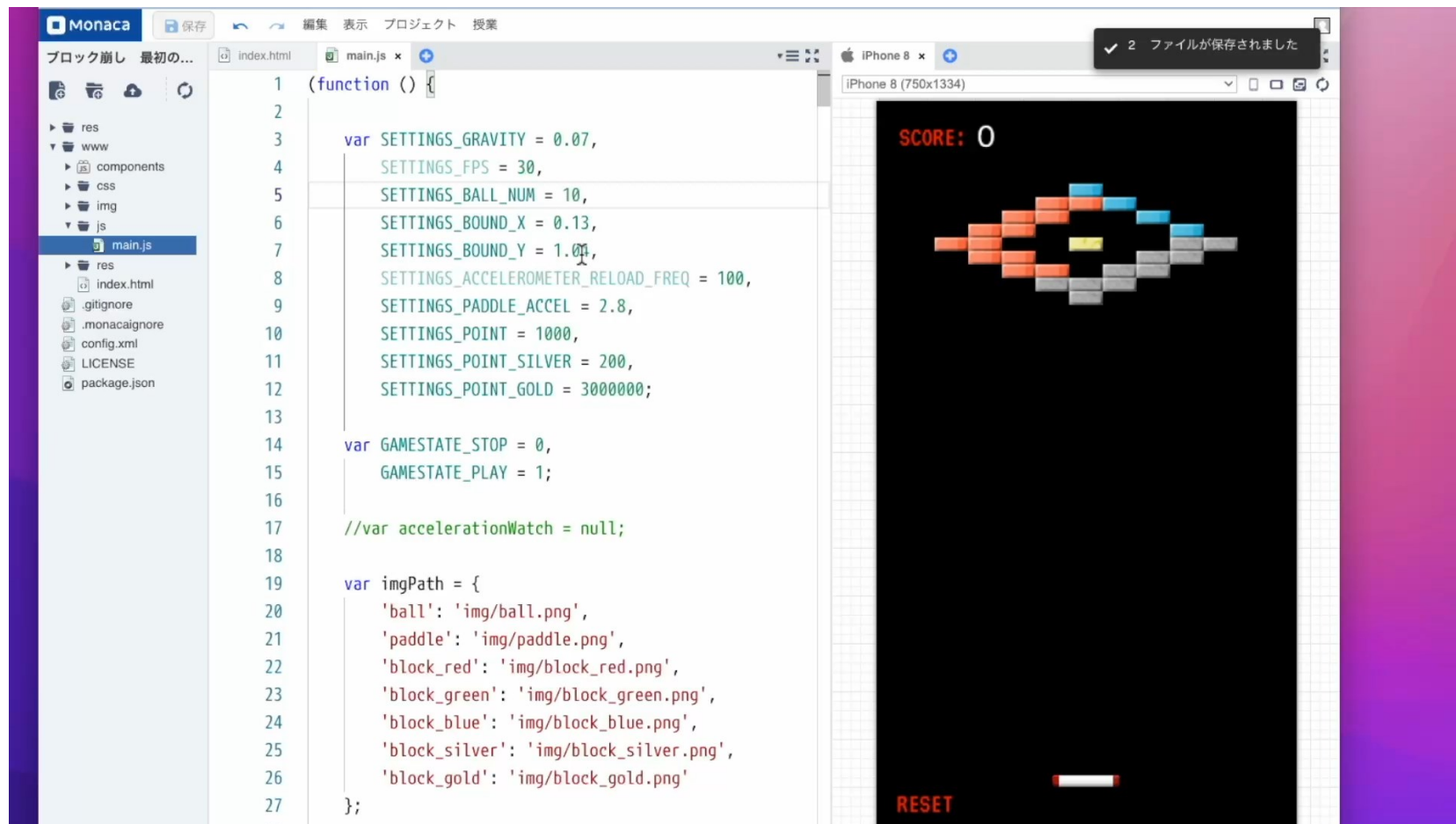
■ 「ブロック崩し」と本資料の位置付け

- 「ブロック崩し」
 - Monaca Educationのプロジェクトテンプレートの一つ
 - HTMLとCSS、JavaScriptで作られている
 - 動作の詳細はライブラリPIXI.jsに委ねている
 - プロジェクトにはソースコードが添付されている
 - ライブラリを利用することで、ソースコードは500行ほど
 - カスタマイズしてゲーム性を変えることができる
- 本資料
 - いくつかのカスタマイズ例を示す
 - 修正量が少なく、難易度が少ないものから、修正量が多く、難易度が高いものへ

ブロック崩しのカスタマイズ例

①最初のボールを増やす

■ ゲーム開始時に落ちてくるボールを増やす



The screenshot displays the Monaca IDE interface. On the left, a file explorer shows the project structure with 'main.js' selected. The central editor shows the following JavaScript code:

```
1 (function () {  
2  
3   var SETTINGS_GRAVITY = 0.07,  
4       SETTINGS_FPS = 30,  
5       SETTINGS_BALL_NUM = 10,  
6       SETTINGS_BOUND_X = 0.13,  
7       SETTINGS_BOUND_Y = 1.04,  
8       SETTINGS_ACCELEROMETER_RELOAD_FREQ = 100,  
9       SETTINGS_PADDLE_ACCEL = 2.8,  
10      SETTINGS_POINT = 1000,  
11      SETTINGS_POINT_SILVER = 200,  
12      SETTINGS_POINT_GOLD = 3000000;  
13  
14      var GAMESTATE_STOP = 0,  
15          GAMESTATE_PLAY = 1;  
16  
17      //var accelerationWatch = null;  
18  
19      var imgPath = {  
20          'ball': 'img/ball.png',  
21          'paddle': 'img/paddle.png',  
22          'block_red': 'img/block_red.png',  
23          'block_green': 'img/block_green.png',  
24          'block_blue': 'img/block_blue.png',  
25          'block_silver': 'img/block_silver.png',  
26          'block_gold': 'img/block_gold.png'  
27      };  
--
```

On the right, the iPhone 8 simulator shows a game screen with a score of 0, a paddle at the bottom, and a cluster of colorful blocks (red, green, blue, silver, gold) in the center. A 'RESET' button is visible at the bottom left of the game area.

■ 修正箇所

- main.jsの5行目
 - 変数 SETTINGS_BALL_NUM = 1 の箇所を

```
ex.html main.js x +
1 (function () {
2
3     var SETTINGS_GRAVITY = 0.07,
4         SETTINGS_FPS = 30,
5         SETTINGS_BALL_NUM = 1,
6         SETTINGS_BOUND_X = 0.12
```

- SETTINGS_BALL_NUM = 10 など、1より大きい**整数**に変えて**保存**

```
4         SETTINGS_FPS = 30,
5         SETTINGS_BALL_NUM = 10,
6         SETTINGS_BOUND_X = 0.12
```

■ 注意事項

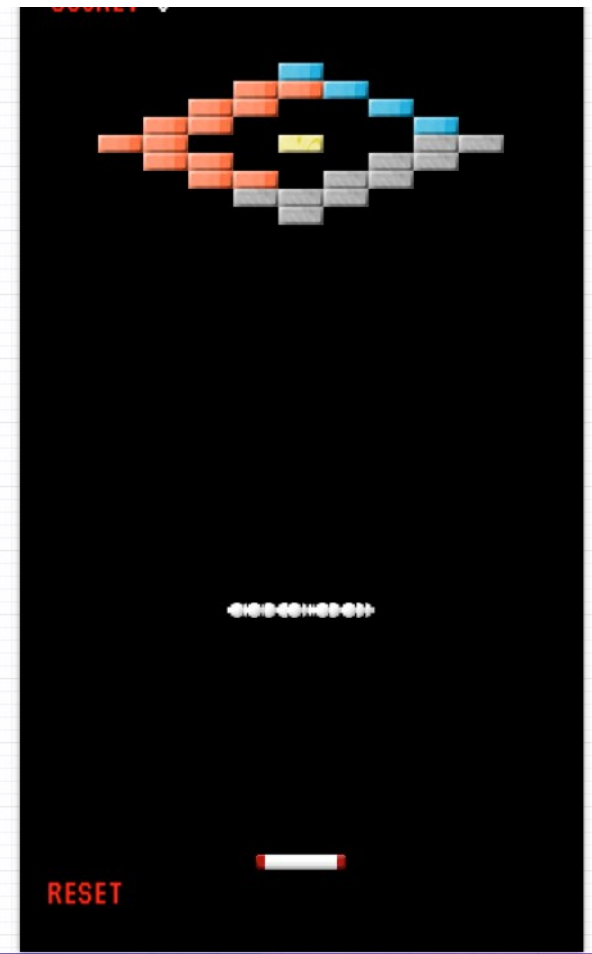
- SETTINGS_BALL_NUMの値は500, 1000などの値にできるが、数が多くなると計算の負荷が高まり、Webブラウザの動作に支障が出るため、数千程度までにする
- Webブラウザの動作がおかしくなった場合、一度ブラウザを閉じてMonaca Educationにログインしなおした後、「セーフモード」でプロジェクトを開いて、数を減らすようにする
 - 参考: セーフモードでプロジェクトを開く
 - <https://edu.monaca.io/function/safemode>

■ 補足事項・学習のヒント：乱数

- ゲーム開始時、ボールは左右方向（X軸方向）に**ランダム**に投げられる
- ボールの数を増やしたとき、それぞれのボールは左右にランダムに落とされる
- そのため、ボールはそれぞれバラバラな方向に落ちる
- main.jsの124行目から127行目

```
124     ball.delta = {  
125         'x': Math.random() - 0.5,  
126         'y': -0.4  
127     };
```

このball.delta.xの値を変更すると、ばらつき度合いが変わる

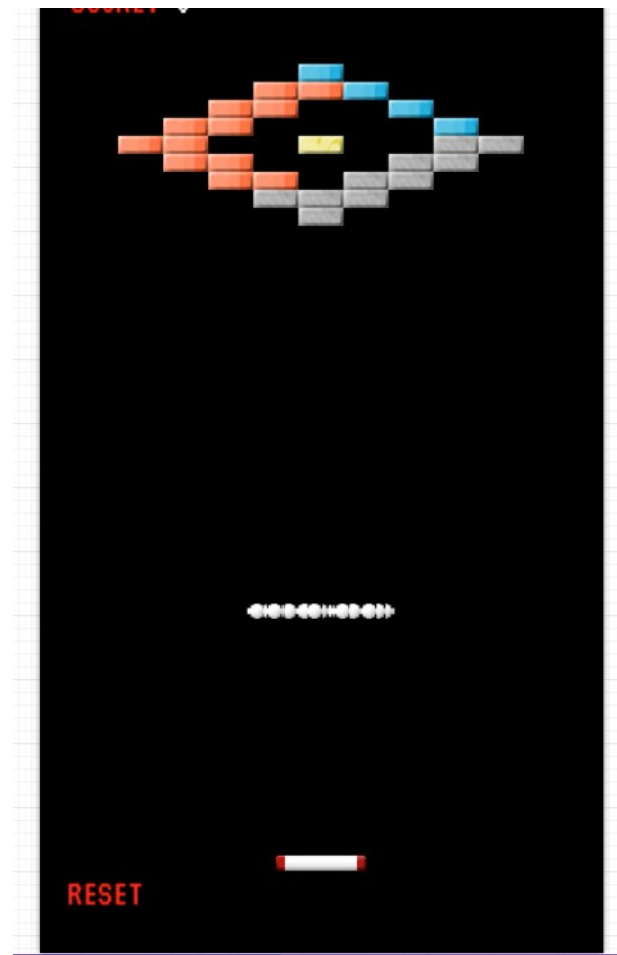


■ 補足事項・学習のヒント：放物運動

- main.jsの124行目から127行目

124			ball.delta = {
125			'x': Math.random() - 0.5,
126			'y': -0.4
127			};

- それぞれのボールには、横方向にバラバラの速度を設定されている
- 縦方向の速度は同じ
- 「放物運動のシミュレーション」とみなすことができる



■ 補足事項・学習のヒント：モンテカルロ法

- たくさんのボールを落とすと、
画面の幅全体に広がる
- そのうち、いくつかがパドルに
当たるかによって、
確率的にパドルの幅が求められる
- モンテカルロ法を学ぶときに、
イメージを沸かせるヒントとして
利用できる

