

ブロック崩しのカスタマイズ例

⑥ブロックに当たったら、ボールを増やす

■ ブロックに当たったら、ボールを増やす

```
375         BB.endGame();
376     }
377     continue;
378 }
379
380 // Ball&Paddle hit detection
381 if (BB.isBallHit(ball, BB.paddle)) {
382     ball.delta.x += -1 * SETTINGS_BOUND_X * (BB.paddle.position.x - ball.
383         position.x);
384     ball.delta.y *= -1 * SETTINGS_BOUND_Y;
385 }
386
387 //Ball&blocks hit detection
388 for (var j = BB.blocks.length - 1; j >= 0; j--) {
389     var block = BB.blocks[j];
390     if (BB.isBallHit(ball, block)) {
391         BB.addBall();
392         BB.addScore(block.point);
393         ball.delta.y *= -1;
394         if ((ball.position.x < 4 + block.position.x - (block.width * 0.5)) ||
395             (ball.position.x > -4 + block.position.x + (block.width * 0.5))) {
396             ball.delta.x *= -1; //ball hits side of the block
397         }
398         BB.stage.removeChild(block);
399         BB.blocks.splice(j, 1);
400         if (BB.blocks.length <= 0) {
401             BB.clearGame();
402         }
403     }
404 }
```

■ 修正箇所

- main.js 386行目～ (animate関数内)
 - 関数animate()
 - 瞬間ごとの、ゲーム画面内の状態について計算・判定する
 - 386行目のコメント `//Ball&blocks hit detection`
 - ボールとブロックのヒット検知
 - 390行目に、以下のコードを追加する

```
386 //Ball&blocks hit detection
387 for (var j = BB.blocks.length - 1; j >= 0; j--) {
388     var block = BB.blocks[j];
389     if (BB.isBallHit(ball, block)) {
390         BB.addBall();
391         BB.addScore(block.point);
392         ball.delta.y *= -1;
```

- BB.isBallHit()でボールがブロックに当たっているか判定し、
当たっていたらaddBall()する