

Flash ゲーム  
「ブロック破壊」

Mr.caffeine

## 目次

1. はじめに
2. Flash とは
  - 2.1 Flash の歴史
  - 2.2 Flash でできること
  - 2.3 Flash の問題点と今後
3. ActionScript とは
4. Flash ゲーム「ブロック崩し」
  - 4.1 計画
  - 4.2 スタート画面
  - 4.3 ゲームプレイ画面
  - 4.4 終了画面
5. おわりに
6. 参考文献

## 1. はじめに

本稿では、Flash 作成ソフト「KoolMoves」を利用して制作した Flash ゲーム「ブロック破壊」について解説する。また、ActionScript を使用し、ゲームを制作する過程やソースコードについても説明する。

## 2. Flash とは

### 2.1 Flash の歴史

Flash とは Adobe 社が提供する、音やグラフィックのアニメーションを組み合わせる Web コンテンツを再生・実行するためのデータ形式の一つであり、それらのコンテンツを作成するソフトウェアのことである。(参照: ホームページファースト <https://www.hp-1st.net/column/0008.html> 2020-01-05 最終アクセス) 当初の開発会社は Macromedia であり、1996 年にアメリカのソフトウェア開発会社であるフューチャーウェーブ・ソフトのソフトウェアを買収してリリースされた。また、その後のアップデートにより、ActionScript が搭載され、プログラミングの機能が強化された。ActionScript については後述する。2005 年に Macromedia は Adobe に買収され、現在の Adobe Flash となった。その後もアップデートを重ね様々な機能が追加されたが、現在は開発終了されており、HTML5 への移行が推奨されている。2020 年に Flash 再生プラグインである Flash Player の開発と配布の終了が予定されており、環境の移行が早急に求められる。

### 2.2 Flash でできること

Flash を用いれば、Web ページを動的に表示することができる。文字に動きを付けたり、ボタンを実装したりすることが簡単に行える。データが軽く無劣化で、色や形が容易に変更できるベクターアニメーションを表現できるため、2000 年代のインターネット黎明期における Web サイトに多く導入された。世界中で人気の動画サイト YouTube も以前は動画プレイヤーに Flash を用いていた。しかし、現在はこちらも HTML5 への切り替えが行われ

ている。

Flash の魅力の本質は動的な表示、つまりアニメーションを付けられることであり、同技術を使用して作られたアニメも多数公開された。その代表的な作品に「秘密結社 鷹の爪」がある。本作品は短期間でベクターアニメーションが作成できる Flash の特徴を活用し、インターネットからテレビへと進出した。通常のデジタルアニメーションと比べて低予算で制作できることが、映像業界からも注目された要因であると考えられる。人気が出ると、スポンサーを数多く獲得し、Flash アニメをして世界初の映画化を果たした。また、関連するゲームも発売され、一定の売上を実現した。

Flash アニメやゲームは個人でも簡単に制作でき、2000 年代後半には多くの人々がニコニコ動画などにアニメを公開したり、個人制作されたゲームを配布したりした。

### 2.3 Flash の問題点と今後

Flash では ActionScript による任意のコードが実行可能であることから、セキュリティ問題が指摘されるようになった。Flash はサンドボックスで実行されるため、直接コンピュータに損害を与えることはないが、PDF 上で Flash を再生できる Adobe Reader には脆弱性が発見されており、危険性があるといえる。

上記にもあるとおり、Flash は 2020 年に開発・配布の終了が予定されており、プラグインが不要である HTML5 への移行が推奨されている。開発が終了された背景には Apple iPhone や iPad の台頭がある。同端末の OS である iOS・iPad OS やそれらの標準ブラウザである Safari は当初から Flash をサポートしておらず、それらの利用者の増加に伴い、新たな環境への対応も進められるようになった。その結果 Flash の利用は減少し、代替技術への移行が進められた。そして各社は Flash を非推奨とし、ついに 2020 年開発・配布が終了する。

それに伴い、Web ページやブラウザゲームのデベロッパーは、環境の移行をしなくてはならない。同時にモバイル向けに最適化されたコンテンツを提供することに注力すべきだと考えられる。

### 3. ActionScript とは

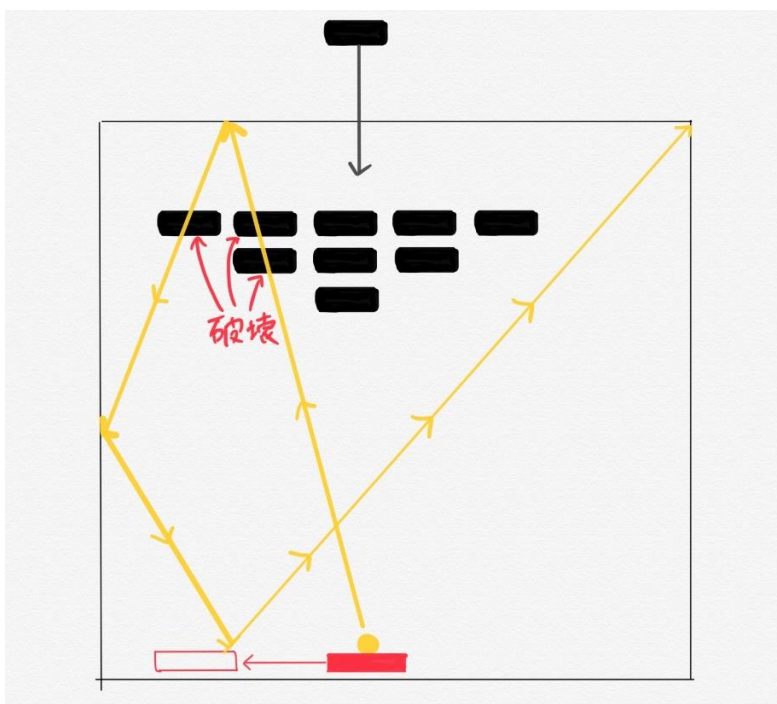
ActionScript とは Flash に使用されるオブジェクト指向のプログラミング言語である。2003 年に使用可能になり、Flash の普及とともに多くの Web サイトで利用されるようになった。日本の大手動画サイト「ニコニコ動画」でも以前は動画プレイヤーが ActionScript で実装されており、ニコニコ動画誕生に不可欠な言語であった。(参照:ニコニコ大百科 <https://dic.nicovideo.jp/a/actionscript> 2020-01-06 最終アクセス)

コードの説明や実装方法は実際のゲーム制作の中で解説する。

### 4. Flash ゲーム「ブロック崩し」

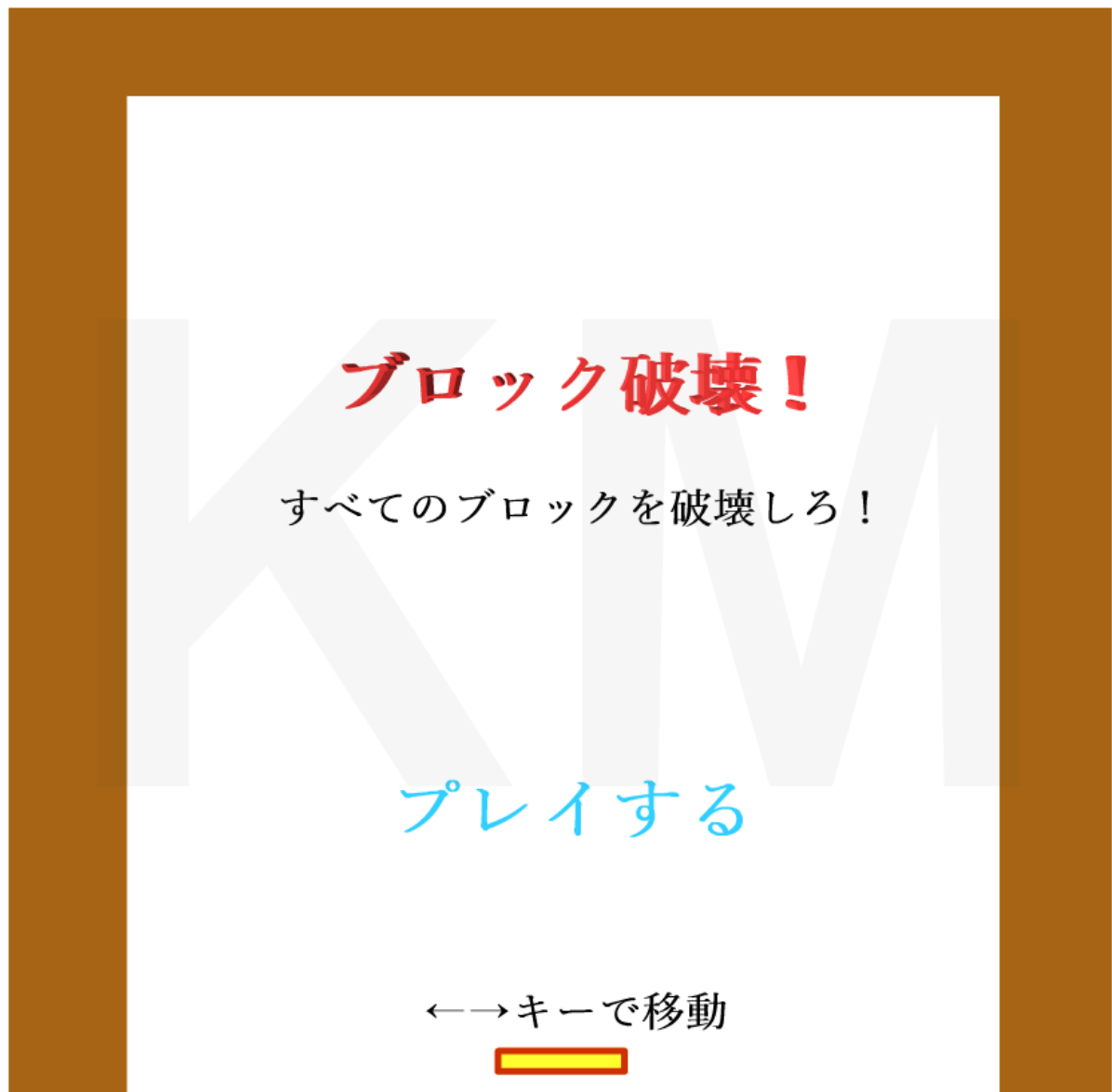
#### 4.1 計画

ここからは実際に Flash・ActionScript を利用し、ゲームを制作する。使用ソフトは KoolMoves Demo である。無料版であるため保存制限があるが、一枚のキーフレームであれば保存できるので、工夫して制作した。



計画したゲームは次のとおりである。ラケットから玉を発射し、玉がブロックを通過するとそのブロックが破壊される。ラケットは矢印キーで左右に動かせ、玉を打ち返せる。玉は壁に当たると反射する。クリア条件はブロックの全破壊であり、ゲームオーバーは下辺にボールが行き、ラケットで打ち返せなくなったときである。また、スタート画面と終了画面を作り、サウンドも実装する。

#### 4.2 スタート画面



スタート画面は画像のようになっている。指定されたリンクをクリックすると専用のHTML ページに遷移し、スタート画面(第2 キーフレーム)が表示される。「プレイする」を押すまでは次の画面に移動しないように、「ムービーの停止」アクションを追加した。第一キーフレームに「ムービーの停止」アクションを追加するとプレイヤーによってはうまく動作しないことがあるため、空の第1 キーフレームを挿入し、Tween を0 にすることで、ただちに第2 キーフレームに移動するようにした。スタート画面では、「ブロック破壊」というタイトルがスピンするようなエフェクトを追加した。

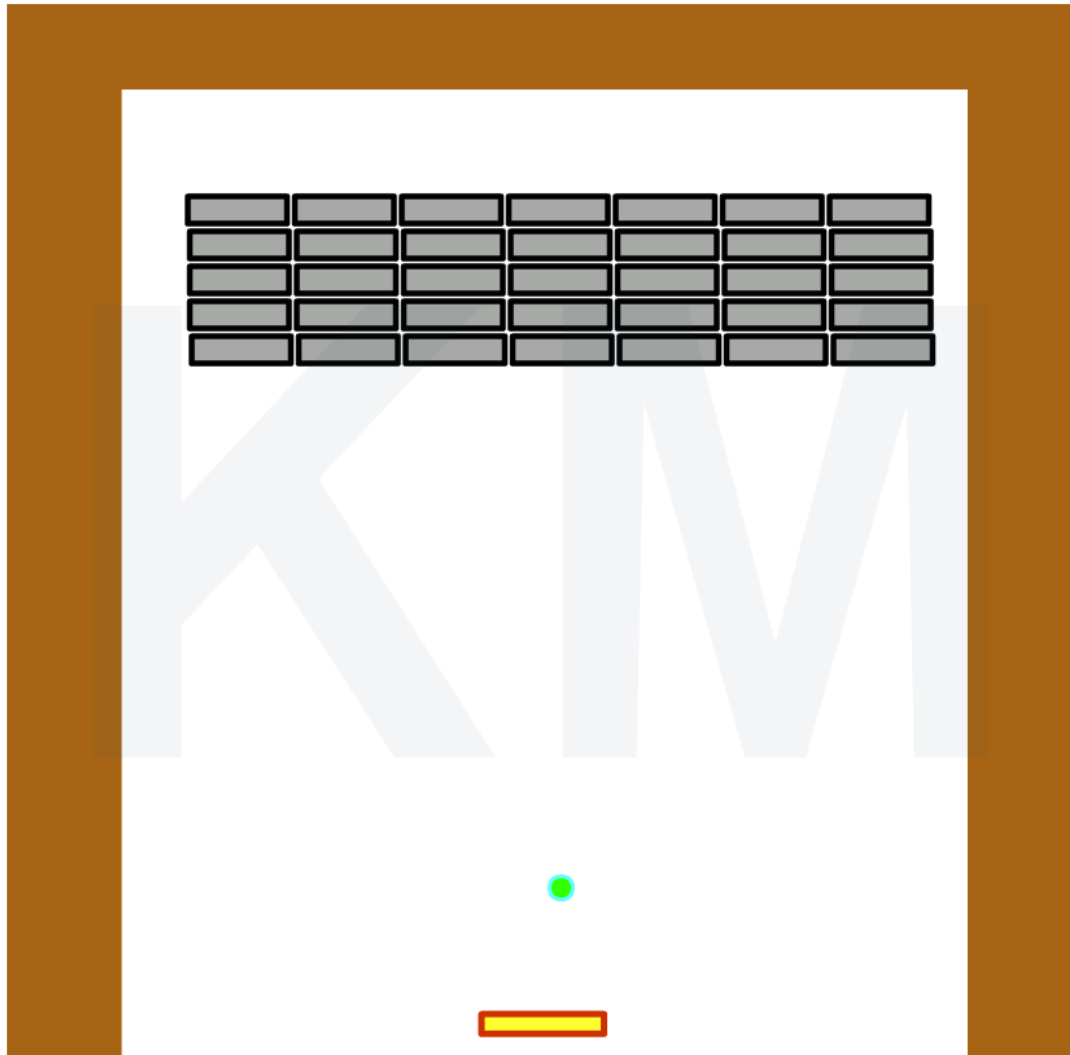


また、矢印キーでの移動が正常に行えるかがゲームプレイ前にわかるように、この状態でも、ラケットの移動ができるようにした。

“プレイする”という文字は押せることがわかるように、ムービー内ムービーを作成して、マウスカーソルが乗っているときだけ少しだけ大きく、色が変わるように実装した。

プレイする

### 4.3 ゲームプレイ画面

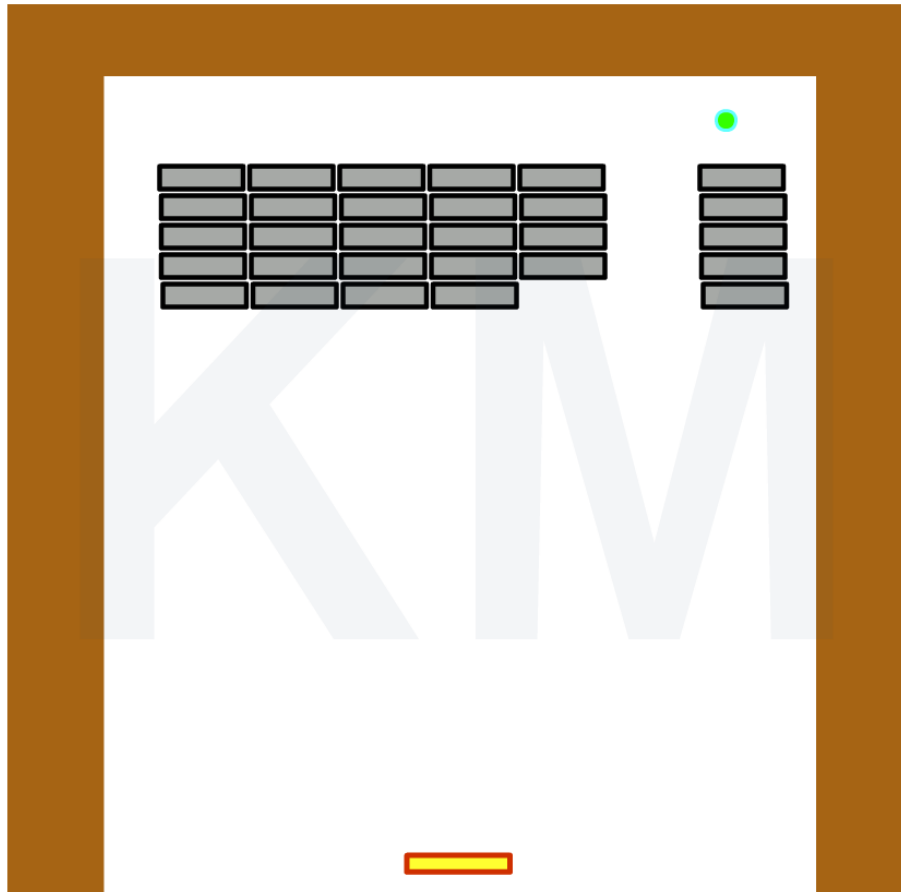


“プレイする”押すとゲームプレイ画面である第3キーフレームに移動し、即座に玉が発射されるようにするため以下のような ActionScript を玉に記述した。

```
onClipEvent (load) {  
    _x = 250.0;  
    _y = 532.0;  
    dx = 7; dy = -20;  
}
```



onClipEvent(load)は当該キーフレームがロードされた時に実行される命令文である。自身の x と y 座標を指定し、初期位置となるよう設定する。dx,dy に値を代入し、移動量を加算する。



次に玉がブロックにあたったときに非表示にしたい。そこで以下の通り記述をした。

```
onClipEvent (enterFrame) {  
    if (_root.mc2.hitTest(_root.block0)) {  
        if (flg0 == 0) {  
            _root.block0._visible = false;  
            flg0 = 1;  
        }  
    }  
}
```

onClipEvent (enterFrame)はフレームが再生されるたびに呼び出される命令文である。玉(mc2)とブロックを hitTest で当たり判定を取る。ここで flg(n)という状態変数(ユーザー変数)作り、初期値を 0 とした。

```
onClipEvent (load) {  
    flg0 = 0;  
}
```

flg が 0 のときに block を非表示にする、つまり visible = false とした。それと同時に flg を 1 にし、一度だけ処理が行われるようにした。

すべてのブロックに固有の flg と block ナンバーを付け、一つずつ処理が行われるようにした。

クリア判定にも flg という変数を使い、すべての flg の値が 1 になったとき(=すべてのブロックが非表示になったとき)にクリア画面である第 5 キーフレームに移動させるようにした。

```
on(enterFrame){  
    if(flgs0==1  
        & flgs1==1  
        & flgs2==1  
        & flgs3==1  
        & flgs4==1  
        & flgs5==1  
        (省略  
        & flgs34==1){  
        _root.gotoAndStop(5);  
    }  
}
```

次にゲームオーバー判定と壁に当たった際の反射を考える。

```
onClipEvent (enterFrame) {  
    _x = _x + dx;  
    _y = _y + dy;  
  
    // 左端に来たとき  
    if (_x < 60 + 5) {  
        _x = 60 + 5;  
        dx = dx * -1;  
    }  
  
    // 右端に来たとき  
    if (_x > 495 - 5) {  
        _x = 495 - 5;  
        dx = dx * -1;  
    }  
  
    // 上端に来たとき  
    if (_y < 46 + 5) {  
        _y = 46 + 5;  
        dy = dy * -1;  
    }  
  
    // 下端に来たとき(ゲームオーバー)  
    if (_y > 535) {  
        _root.gotoAndStop(4);  
    }  
}
```

上記のように、それぞれの壁が位置する座標に到達したときに、移動方向を変えるという記述を行った。ラケットよりも下の座標に玉が到達したときは、第4キーフレーム(ゲームオーバー画面)へと移動させ、ゲームが終了する。

次に、ラケットに関する記述である。

```
onClipEvent (load) {
    _x = 275;
    _y = 525;
}
onClipEvent (enterFrame) {
    if (Key.isDown(Key.LEFT)) {
        _x = _x - 16;
    }
    if (Key.isDown(Key.RIGHT)) {
        _x = _x + 16;
    }
    if (_x < 0 + 90) {
        _x = 0 + 90;
    }
    if (_x > 480 - 20) {
        _x = 480 - 20;
    }
}
```

こちらにも玉同様に初期位置の座標を指定した。一定の範囲を矢印キーで左右に移動するよう記述を行った。

```
if (_root.mc2.hitTest(_root.mc1)) {
    _root.mc2._y = _root.mc1._y - 10;
    // ラケットの左端に当たったとき
```

```

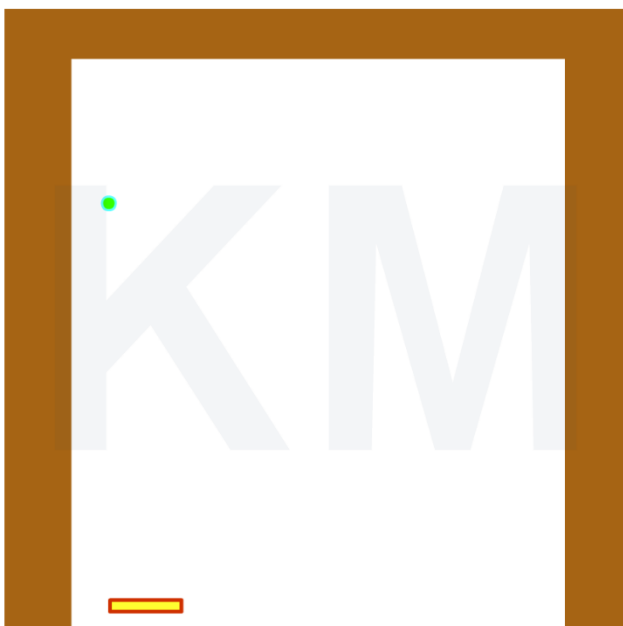
if(_root.mc1._x + 20 < _root.mc2._x){
    dx = 3;
    dy *= -1;
// ラケットの右端に当たったとき
}else if(_root.mc2._x + 20 < _root.mc1._x){
    dx = -3;
    dy *= -1;
}else{
    dy *= -1;
}

```

ラケットの右側に当たったときには右に、左側に当たったときには左に移動するように工夫した。

#### 4.4 終了画面

すべてのブロックが破壊され尽くしゲームクリアになる、または、下辺に玉が落ちゲームオーバーになると終了画面へと移動する。



～クリア画面～



～ゲームオーバー画面～



終了画面ではどちらも、“もう一度プレイする”ボタンを設置し、何度でも挑戦できる仕様とした。また、キーフレームに移動した際にサウンドが出るようにした。

以上でゲームは一通り完成である。

## 5. おわりに

BGM や効果音を多くつける予定であったが、ファイルサイズが大きくなると再生できなくなり、クラッシュも多発するようになり、断念せざるを得なかった。また、ブロック自体が動くようにしたかったが、こちらも動作が重くなり、実現することはできなかった。このように Flash でのゲーム制作は難点が数多く存在し、快適に開発ができる環境とはもはや言えなかったという現状がある。

冒頭にも記載した通り、Flash は 2020 年に開発が終了され、転機が訪れる。今後は他のゲームエンジンへの移行をすすめるなどして、さらに高度なゲーム制作に挑戦したい。

## 6. 参考文献

- ・ ホームページファースト

<https://www.hp-1st.net/column/0008.html> 2020-01-05 最終アクセス

- ・ ニコニコ大百科

<https://dic.nicovideo.jp/a/actionscript> 2020-01-06 最終アクセス

- ・ Flash ゲーム講座・アクションスクリプトサンプル集

[https://hakuhin.jp/as/block.html#BLOCK\\_01](https://hakuhin.jp/as/block.html#BLOCK_01) 2020-01-06 最終アクセス

- ・ 旭 貴朗 Flash プログラミング

<http://www2.toyo.ac.jp/~asahi/education/soron/shiryo/application/picture/flash/index.html> 2020-01-06 最終アクセス