

カビパン

Creator
タナカミノル (pickles)

デジタルでリアルを表現する

ムービーの中でカビが増殖していくコンテンツ。
ポイントは、カビの増殖をどうスクリプトで表現するか。



Title
カビパン

Sample URL
<http://www.pickles.tv/dotfla/01/>

Archive
kabipan.zip

File
09

スクリプト
ActionScript 1.0

対応プレーヤー
Flash Player 6以上

制作アプリケーション
Flash MX

発案～デザイン

グラフィックをもとに表現方法を検討する

この作品を作ろうと思ったのは、「デジタルでデジタルなものを表現してもおもしろくない」というのが起点になっています。じゃあデジタルでリアルなもの、かつ、誰もやったことないもの…と考えて、「カビなら他の人がやっているのを見たことがないし、そんなにむずかしくないのでは？」と思い、チャレンジしてみることにしました。

○ グラフィック作り

まずは、イメージの確認のためにグラフィックを作ります。カビを繁殖させる媒体は、生活に密着していて一番目につくものということで、食パンにしました。最初は1枚で考えていたのですが、キャンバスとして広く使えるので2枚にしました。

最初のグラフィックイメージ



できあがったグラフィックを見て、検討します。カビっぽいグラフィック断片を複数用意して増殖させようと思いましたが、それでは処理が重くなってしまいます。そこで、ベクターでの表現を検討することになりました。

グラフィックイメージを拡大してみたところ、小さな十字形の集合体で表現できるのではないかと思い、十字形のオブジェクトを増殖させてカビを表現することになりました。

カビ部分の拡大図

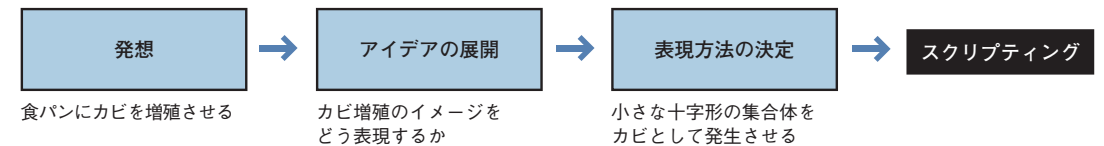


スクリプト

カビの特性をスクリプト化する

あとはスクリプトを組んでみて、作り込んでいきます。

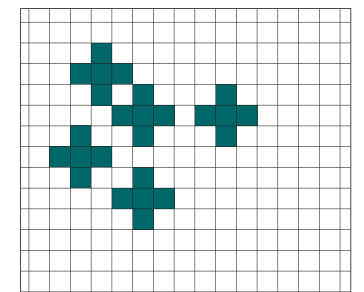
スクリプトまでの流れ



○ 最初のスクリプト

最初に作成したスクリプトは以下の形です。3×3 (ピクセル) の十字形の MC を作成し、リンケージで "kabi" という名前を付け、それをランダムに配置していく形です。

ランダムに配置



SOURCE 初期コード

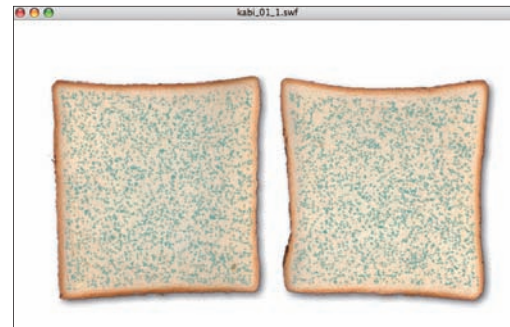
```

// 初期設定
var i = 0;
// カビオブジェクトの ID
// 増殖ルーチン
onEnterFrame = function () {
  // カビの生成される場所
  var xpos = Math.floor(Math.random()*750);
  var ypos = Math.floor(Math.random()*500);
  // カビがパンの上当たったらカビオブジェクトを生成
  if (_root.pan.hitTest(xpos, ypos, true)) {
    i++;
    var newname = "kabi_"+i;
    this.attachMovie("kabi", newname, i);
    this[newname]._x = xpos;
    this[newname]._y = ypos;
  }
};
  
```

結果、右図のように増殖すると均一化されてしまい、実際のカビの表現とはほど遠いものになってしまいました。

これを解消するために、カビとその特性について考えてみることにします。

増殖は均一化



●カビの特性を考える

スクリプトの均一化した動きはカビっぽくありません。では、カビらしい動きとはどんなものなのでしょうか？

- ・隣接しているところから増殖
- ・たまに離れたところから増殖
- ・もともとある場所にも増殖

この3つの要素を満たしたルーチンを作れば、たぶんカビになりそうです。

○2 番目のスクリプト

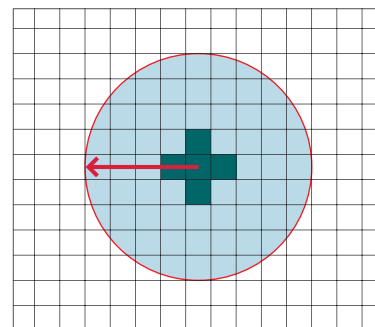
●「隣接しているところから増殖」と「たまに離れたところから増殖」

まずは均等配置にならないように、生成された近くに新しく生成されるというルーチンを考えました。最初に配置された場所から、半径5の円周上に次のカビがランダムに生成されるようにします。

ただこのままだと、過去に生成された場所にも新しく生成されてしまうので、空のMC内にカビを生成するようにして、自分自身のMCにぶつかった場合は、別のところにオブジェクトが生成されるようにしました。

これによりカビ同士の重なりがなくなり、確実に周りに広がっていく形で増殖するはずです。

移動先を半径5の円周上とする



📄 SOURCE 2 番目のコード

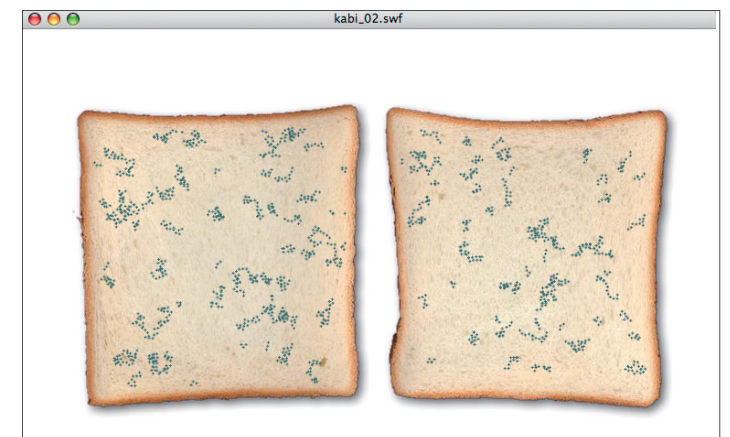
```
// 初期設定
var i = 0; // 複製するカビの ID
var idouhankei = 5; // カビが複製される場所の移動半径
var xpos;
var ypos;
```

👉 次のページに続く

```
var newname;
// ランダムに生成される場所を設定する関数
function random() {
    xpos = Math.floor(Math.random()*750);
    ypos = Math.floor(Math.random()*500);
}
// 増殖ルーチン
random();
onEnterFrame = function () {
    // 新しい生成位置を設定
    k = Math.floor(Math.random()*360); // 角度を設定
    red = k*Math.PI/180; // ラジアンに変更して
    newXpos = xpos+idouhankei*Math.cos(red);
    newYpos = ypos+idouhankei*Math.sin(red);
    if (_parent.pan.hitTest(newXpos, newYpos, true) == true) {
        // もしパンの上に生成され
        if (this.hitTest(newXpos, newYpos, true) == false) {
            // かつ自分自身にぶつからなかった場合は、カビを生成して配置する
            i++;
            newname = "kabi_" + i;
            this.attachMovie("kabi", newname, i);
            this[newname]._x = newXpos;
            this[newname]._y = newYpos;
            // 新しい生成位置を現在の位置に設定
            xpos = newXpos;
            ypos = newYpos;
        } else {
            random();
        }
    } else {
        random();
    }
};
```

このスクリプトで生成されたものが右図になるのですが、によりよると生成された形で、思いどおりのイメージにはほど遠いものでした。

によりよるとした増殖



●「もともとある場所にも増殖」を追加

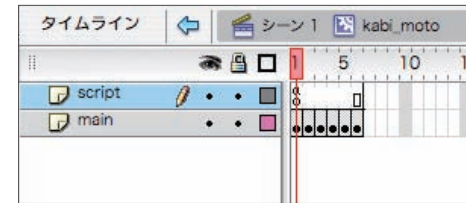
そこで、さらに変更を加えます。自分の MC にぶつかった場合には、過去に生成したカビオブジェクトを大きくするようにしました。先にあげたカビの特性のうち「もともとある場所にも増殖」部分の追加です。

新しく生成しつつ、過去生成したカビも大きくなっていく形です。ここは非常にローテクですが、下図の 6 フレームのアニメで再現しています。

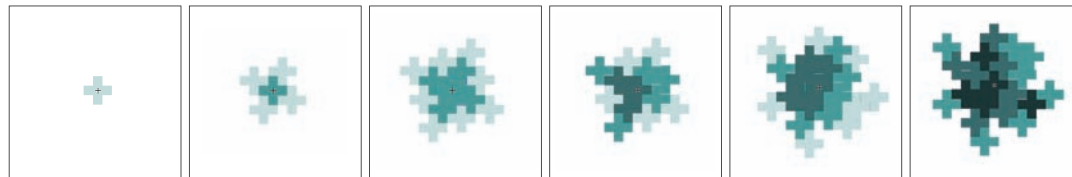
自分自身の MC にぶつかった場合は、このフレームを進めることで「もともとある場所にも増殖」を表現します。また、1 フレーム目はもともと濃いグリーンでしたが、透明度を 30% まで落として、視認しにくくしています。

これにより、による増殖している感じが払拭されると考えました。

「もともとある場所にも増殖」はアニメーションで再現



カビの特性を表現するためのアニメーション (フレーム 1～6)



スクリプトは、「2番目のコード」(98 ページ) の増殖ルーチンのうち **A** の部分を変更して、以下のようになりました。

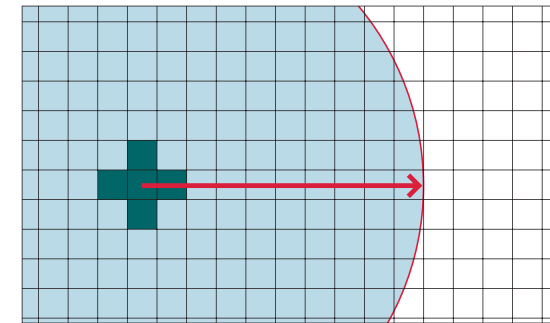
SOURCE 変更後の増殖ルーチン

```
var kakudo = Math.floor(Math.random()*4);
this[newname]._rotation = 90*kakudo;
idouhankei = 5;
}else{
// このカビ MC 上に生成されてしまったら、今まで生成したカビから
// ランダムで選び出して
var ran = Math.floor(Math.random()*i);
newname = "kabi_"+ran;
// そのカビを次のフレームに進める
this[newname].nextFrame();
xpos = this[newname]._x;
ypos = this[newname]._y;
// そのカビの場所を次に生成する場所の原点にして
idouhankei = 10; B
// 移動半径を倍の数にする
```

ここでのポイントは、回転させ変化を付けていることです。回転を付けることによって、まったく同じグラフィックが増殖していることに気付かれないようにしています。

また **B** の部分で、移動先の半径を大きくしています。初期に設定している半径 5 では、アニメの 6 フレーム目だったりと、自身の MC にかかなりの確率で当たってしまうので、近辺に新たなカビオブジェクトを生成させるために半径を大きく取っています。

移動先を半径 10 の円周上とする



これで、一連の流れができあがりました。実行すると、リアルな表現で食パンにカビが増殖していきます。

食パンにカビが増殖



なお、無限に増殖しないように、ラストに以下のスクリプトを追加しています。ここで 2500 としているのは、MC が 2500 以上生成されると極端に動作速度が落ちるためです。

SOURCE 無限増殖を禁止

```
if(i>2500){
delete onEnterFrame;
}
```