

状態遷移図に基づいた Flash Animation 記述環境

中川昌幸 (茨城大)・岡本秀輔 (成蹊大)・鎌田賢・米倉達広 (茨城大)

1 はじめに

Web 上で提供されているオンラインゲームやインタラクティブアニメーションの多くは、Macromedia Flash でかかれている。

Macromedia Flash MX などの Flash 制作ツールでインタラクティブなコンテンツを作成しようとすると、図柄のデザインのみならず、ActionScript を駆使したプログラミングが必要である。Flash Maker などのツールは予め決められたアクションパターンの組み合わせとしてアニメーションを作成できる。ユーザは ActionScript を記述しなくてもよいが、複雑な動きを表そうとしたりインタラクティブアニメーションを作ろうとすると複雑化してしまう。

対話型アニメーション作成ツールである Islay では、紙の上で設計した、あるいは頭で思い浮かべた状態遷移図を直接的、直観的にアニメーションにできる初心者に優しいユーザインタフェースを提供している。プログラミング言語である Sueak[3] や、図形書き換えルールに基づく Viscnit[4] と比べて、古典的な状態遷移図だけで記述する事が Islay の特徴である。本稿では Islay で記述されたアニメーションを Flash 形式で出力するトランスレータを実装することにより、状態遷移図に基づく Flash Animation の記述環境を実現する。

2 状態遷移図とムービークリップとの対応

図 1 は、Islay でのアニメーション定義の例である。1 つの丸は、1 つの状態を表し、表示画像と移動などのアクティビティに対応している。1 つの矢印は、状態から他の状態への遷移を表し、一般には衝突などのイベントが発生したときに発動される。

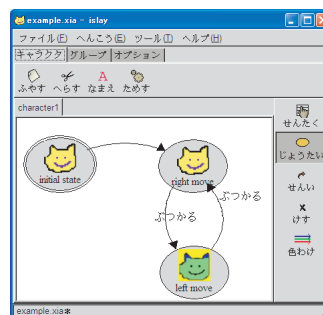


図 1 Islay でのアニメーション定義の例

一般に Flash ムービーは、ムービークリップやボタンなどのシンボルの集合として構成される。本稿の方式では、1 つのキャラクタを定義する状態遷移図を 1 種類のムービークリップに対応づける。その動作は、アクションスクリプトによって記述される。このアクションスクリプトにおいて、各状態は、メソッドとして表現される。各スクリプトには関数ポインタがあり、それがどのメソッドを指しているかによって、そのキャラクタがどの状態にあるのかを表す。状態の遷移は、メソッドが関数ポインタの指し先を他のメソッドに書き換えることによって実現される。このようなムービークリップを状態の数だけ生成し、それぞれの状態に対応する表示される画像を持たせる。状態の数と同じだけの多数のムービークリップが同時に同じ位置に存在することになるが、各ムービークリップが、表示画像に対応する状態にあるとき以外は非表示になるようにすれば、表示画像は 1 つに限られて、あたかも 1 つのキャラクタが表示画像を変えながら動いているように見える。

以上によって、状態遷移図で記述される Islay のアニメーションは、Flash 形式のアニメーションに変換される。

```

example.ia
1  trans dgm0 {
2    s0 (0, jump(0, 0)) {
3      default :s1;
4    }
5    s1 (0, move(10,0)) {
6      bump : s2;
7      default :s1;
8    }
9    s2 (0, move(-10,0)) {
10     bump : s1;
11     default :s2;
12   }
13 }
14 };

```

図 2 中間言語表現の例

3 具体例

図 1 の状態遷移図は、衝突したらキャラクターの表示画像と移動の向きを変えるという単純なアニメーションを定義している。図 2 は、その中間言語表現である。図 2 において、例えば *s0* は、初期状態を表し、そこでのアクティビティが座標 (0,0) へのジャンプであること、無条件で状態 *s1* へ遷移することを表している。また、*s1* は (10,0) 方向への移動をアクティビティとしてもち、何かに衝突したら状態 *s2* へ遷移することを表している。

図 3 は、状態 *s1* に対応するムービークリップに貼り付ける ActionScript である。関数ポインタ *func* は初期状態 *s0* を指すように初期化される (22 行)。*onEnterFrame* 関数はフレームが 1 つ進むたびに実行される関数であり、状態のアクティビティを実行 (24 行) したあと、遷移が起きた場合は *func* が上書きされる (25 行)。*function s2* 内の *return s1* は状態 *s1* に遷移することを意味しているが、その前にムービークリップを *visible* にして (18 行) 状態 *s1* に対応する画像が表示されるようにしている。

実際に Flash の swf ファイルを出力し、それを実行した例を図 4 に示す。

参考文献

[1] Toshio Morioka "A tool for authoring interactive animation based on state transition diagram", Master's thesis, Graduate School of

```

example.as
1  function s0(flag) {
2    ...
3  };
4  function s1(flag) {
5    if (flag) {
6      move(10, 0);
7    } else {
8      if (myHitTest()) {
9        this._visible = false;
10       return s2;
11     } else {
12       return s1;
13     }
14   }
15 };
16 function s2(flag) {
17   ...
18   this._visible = true;
19   return s1;
20   ...
21 };
22 var func = s0;
23 this.onEnterFrame = function() {
24   this.func(true);
25   this.func = this.func(false);
26 };

```

図 3 ActionScript の例 (状態 *s1*)

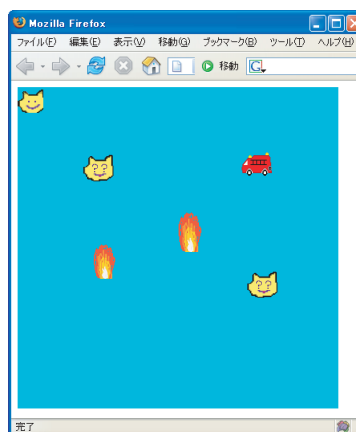


図 4 Flash による実行例

Science and Engineering, Ibaraki University, Feb. 2002.

[2] 岡本 秀輔, 鎌田 賢, 中尾 隆司: 状態遷移図にもとづく対話型アニメーション作成ツールの提案 情報処理学会論文誌: プログラミング, Vol.46, No.SIG 1(PRO 24), pp.19-27, 2005 年 1 月.
[3] Squeak: <http://squeak.org/>.
[4] 原田康徳: <http://www.viscuit.com/>.