

対話型アニメーションにおけるアクタ動作のビジュアル記述方式

4H-8

中川 滋雄

石田 博文

NEC C&C 研究所

1 はじめに

近年、3次元コンピュータグラフィックス(CG)のハードウェアが急激に低価格化し、3次元ゲームなどや建築ウォークスルーなどの「インタラクティブな3次元CG」が身近な存在となってきている。また、Internet上の3次元データ記述仕様であるVRML(Virtual Reality Modeling Language)の普及により、対話型の3次元CGアプリケーションの利用および作成の機会が今後増加すると見込まれる。

筆者らはこのような背景に基づき、大規模な対話型3次元CGアニメーションのシナリオを記述するために、複数のアニメーションシーンをリンクしてシナリオを構築する方式[1]、および、シーン中の登場物体(アクタ)の動作を基本動作のリストによって記述する方式[2]についてすでに報告した。本稿では、アイコンを用いたビジュアルなプログラミングによるアクタの動作手順の記述/編集方式について報告する。

2 テキスト型プログラミング

対話型アニメーションでは、アクタの動作データ(空間中の動きなど)をあらかじめ定義しておき、ユーザの入力などに応じてある一連の手順に基づきながら、これらの動作を変更させる処理を行なう。従来、アクタの動作手順(behavior)の記述に際しては、C言語などのテキスト型のプログラミング言語を利用する必要があった[3]。このため、以下の問題が生じていた。

- 作成者にテキストプログラミングのスキルが必要
- 動作手順が複雑な場合、動作データとアクタの内部状態との対応関係の把握が困難

これらの問題はプログラミングスキルをもたないエンドユーザに対して深刻であるばかりでなく、スキルをもつプログラマにとっても大規模な対話型アニメーション作成時の工数の増大の原因となる。

3 方針

前述の問題を解決するため、以下の特徴をもつ対話型アニメーション作成環境の実現を図る。

Visual Behavior Programming for Interactive 3D Animation,
Shigeo NAKAGAWA, Hirofumi ISHIDA,
C&C Research Laboratories, NEC Corporation

- アクタの動作手順をビジュアルに設定するための編集方式およびユーザインタフェース
- 上記編集方式に適したアクタの動作手順記述のためのデータモデル

上述の動作手順記述のためのデータモデルとしてはすでに報告した動作単位リスト[2]に基づくモデルを採用する。以下、データモデル、続いてビジュアル編集環境の順に説明する。

4 動作手順記述データモデル

アクタの動作データとその開始条件の組み合わせで動作手順の基本データである動作単位を作成し、これを動作の実行順に接続して、一連の動作手順(逐次実行、条件分岐、ループなど)を記述する(動作単位リスト)。動作単位を構成する開始条件および動作データの例として以下がある。

開始条件 ユーザ入力発生、動きデータ終了イベント発生、メッセージ受信

動作データ アクタ動き(空間中での並行移動/回転/拡大縮小)、外部システムコマンド起動、メッセージ送信

5 ビジュアル編集環境

本ビジュアル編集環境では動作単位をアイコンとして扱い、アイコンの接続操作によって動作手順データを作成/編集する。本環境は以下の3種の機能モジュールから構成する。

動作単位作成モジュール

動作データ(他のアニメーションツールなどで作成したアクタの軌跡データなど)に対して、ユーザが開始条件をメニュー選択方式で対応させて動作単位を作成する。アクタの動きデータ(例: CRTの移動)を図1に示す。作成した動作単位は以降アイコンとして扱う。

動作単位アイコン編集モジュール

動作単位アイコンを接続してアクタの動作単位リストを作成/編集する(図2)。

図2では、

- アクタが開始位置 (INIT)
- クリック (MB1) により GO という動きデータ実行
- 直前動き終了 (FIN.) により停止位置 (STOP)
- クリック (MB1) により RETURN という動きデータ実行
- 直前動き終了 (FIN.) により INIT へジャンプ

というループ型の動作手順を記述している。対話型アニメーションの実行画面 (図3) を示す。

上の例では INIT または STOP 実行中にクリック (MB2) により外部システムコマンド (説明画像表示) を起動するよう条件分岐を設定している。

動作単位 DB モジュール

動作単位または動作単位リストをデータベース化して格納する。前述の編集モジュールと連携して作成した動作単位リストをテンプレート化して再利用する。

6 評価および考察

本ビジュアル編集環境を用いて利用評価を行なった結果、本環境を用いずにテキストエディタによって対話型アプリケーションの動作単位リストデータを作成する場合に比較して大幅に作成工数を削減できた (例: アクタ数 20 個のシーンで 1~2 人日程度、テキストエディタ利用時の 1/2~1/3)。

今後の課題として、作成した動きデータと表示アイコンとの関連をより把握しやすくすることなどが挙げられる。

7 おわりに

本稿ではアクタの動作単位を表すアイコンをビジュアルプログラミング環境を用いて記述/編集する方式について述べた。提案方式では、従来のテキスト編集よりも少ない工数で、同時にエンドユーザにも容易な対話型アニメーションの作成手法を提供できる。

参考文献

- [1] 中川、石田、「対話型アニメーションのシナリオデータモデル」、3R-7、第 49 回情報大全
- [2] 中川、石田、「対話型アニメーションのキャラクタ動作記述方式」、4C-9、第 50 回情報大全
- [3] Reynolds, C. W., "Computer Animation with Scripts and Actors", Computer

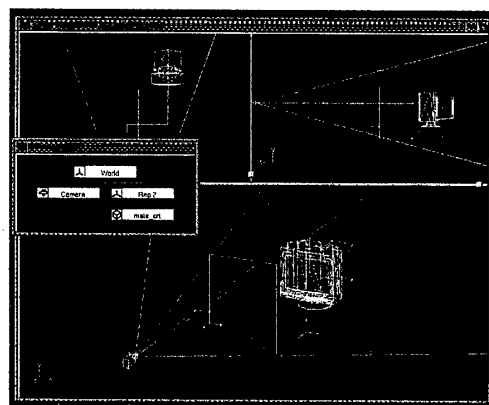


図 1: CRT 動きデータ

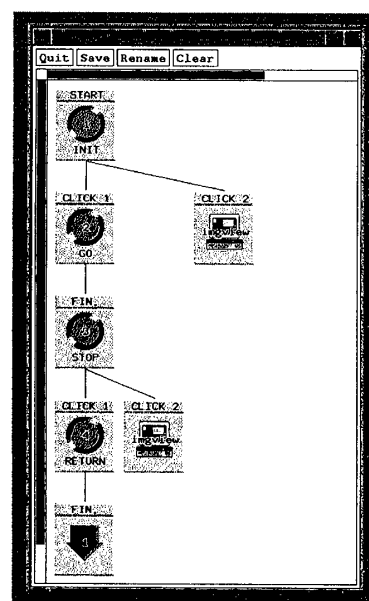


図 2: アイコン編集モジュール



図 3: 対話型 3 次元 CG アニメーション実行画面