

Tablescope Animation におけるキャラクター制作ソフトウェアの開発

山岡 潤一¹ 赤塚 大典² 笈 康明^{1,3} 苗村 健⁴

¹慶應義塾大学 環境情報学部 ²慶應義塾大学 大学院政策・メディア研究科

³科学技術振興機構 さきがけ ⁴東京大学 大学院情報理工学系研究科

1 はじめに

近年のメディア技術の進歩や、インターネットをはじめとするネットワークインフラの整備に伴い、一般のユーザがコンテンツを使うだけでなく、作って発信する時代が到来したと言える。その中で、誰でも簡単にデジタル表現できる環境が強く求められている。このような背景の中、筆者らは、アニメーション制作を支援する環境として Tablescope Animation を開発してきた[1]。

Tablescope Animation とは、卓上の実オブジェクトに対して手描きのキャラクターがリアルタイムに投影されるテーブル型のアニメーション制作環境である(図 1)。ユーザは、卓上オブジェクトを手で動かして操作することにより、キャラクターに動きを付加することができる。そのオブジェクトの動きデータは、バーチャル空間に配置されたキャラクターに反映され、最終的に通常のモニターでも鑑賞可能なアニメーション作品が生成される。

筆者らはこれまで子供向けワークショップなどを通して、本システムを用いた制作体験を提供してきたが、その中でキャラクターの絵を切り替えるルール付けを柔軟に設定出来ないという点や、設定したルールが直感的に分かりづらいという意見があった。今回は、こうした課題の解消に向けて、描画・操作ソフトウェアの改良を提案する。

2 関連研究

ユーザが簡単にかつ複雑な作品を制作可能にするためのアニメーション制作ソフトウェアは、これまでも数多く研究されてきた。代表的な例として五十嵐らの研究[2]では、描画したキャラクターの手足を動かすことでアニメーションを生成できる。また、Modulobe[3]は物理シミュレーションを用い、関節部品を組み合わせることで自由度の高いキャラクター生成を実現している。Video Puppetry[4]はキャラクターを描画した紙をカメラの前で動かすことでアニメーションを生成する。

また、プログラミングの概念を適用してアニメー

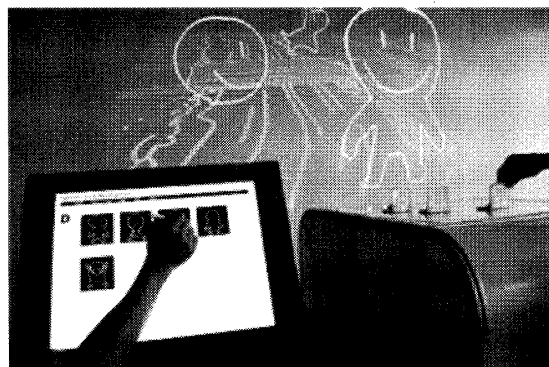


図 1: Tablescope Animation

ションの動き付けを行う試みもいくつかなされてきた。例として、Viscuit [5]は、子供向けのビジュアルプログラミング言語である。描画した図形の配置を変更することで、アニメーションを生成する。

Scratch[6]は、命令文を置き換えたブロックをユーザが自由に並べることでアニメーションを生成する。

このような従来のアニメーション制作ソフトウェアおよびビジュアルプログラミング環境は、モニターの中で完結する制作手法に特化したものであった。一方で、筆者らの Tablescope Animation では、タンジブルインタフェースを用いた独自のルール設定方法やテーブル上の実オブジェクトと対応したビジュアルプログラミングが要求される。

3 キャラクター制作ソフトウェア

3.1 システム概要

上述の通り Tablescope Animation では、ユーザの描いたキャラクターがテーブル上の実オブジェクトに表示され、オブジェクトを実際に手で触れ操作しながら動きを付けることができる。今回は、この中のキャラクター生成部におけるソフトウェア環境をターゲットとする。

このキャラクター生成フェーズでは、ユーザは主に以下の3つの操作を行う。

- キャラクターの絵を描く
- キャラクターの絵の切り替えのルールを作成する
- キャラクターの絵と切り替えルールの対応付けを行う

本システムでは、実オブジェクトの動きや位置、時刻などのルールに応じて事前に描いたキャラクター

Junichi Yamaoka, Daisuke Akatsuka, Yasuaki Kakehi, Takeshi Naemura "Development of a Software for Creating Characters in Tablescope Animation"

1. Keio University, Faculty of Environment and Information Studies
2. Keio University, Graduate School of Media and Governance
3. Presto, Japan Science and Technology Agency
4. Graduate School of Inform. Science and Tech., The Univ. of Tokyo

の絵を差し替えることでアニメーションを表現していく。今回のキャラクタ生成ソフトウェアでは、直感的に画像の切り替えルールを制定するために、図 2 のようにキャラクタ絵を線でつなぐことでルールを設定するグラフィカルプログラミングの方式を採用。ルール設定後、テーブル上の卓上オブジェクトが設定と同じ動きをすれば、対応するキャラクタ絵に切り替わる。卓上オブジェクトに投影されたキャラクタ絵は、ルールに応じて切り替わり、最終的に動画ファイルとして出力される。

3.2 システムの実装

図 2 に実装したソフトウェアの画面デザインを示す。図 2 a のツールバーでは、キャラクタの描画等のモードを切り替えることができる。画面中央はルール設定画面およびキャラクタ絵の状態遷移図である (図 2 b)。また、画面右側 (図 2 c) にてキャラクタの描画を行うことで新たな絵を追加する。図 2 d の画面では、テーブル上の実オブジェクトの動きを画面上でも確認することができる。

本ソフトウェアでは、キャラクタ絵同士を線で繋ぐことでキャラクタの動作パターンを定義する。線はデフォルトのキャラクタ絵からの状態変化を示しており、線上にルールを記述したアイコンを配置することでルールを設定する。

ルールは、プリセットルールとユーザが独自に定めるカスタマイズルールの二種類を用意した。プリセットルールとは、“右へ動く”、“他のオブジェクトと隣り合う”など予め決められたルールセットである。また、プリセットルール以外の柔軟な動きを設定するために、卓上オブジェクトの動きを記録しルールをカスタマイズすることもできる。カスタマイズルールの生成には、卓上オブジェクトをユーザが動かす、その ID および座標をデータセットとして保存する。カスタマイズルールを適用した場合には、テーブル上の実オブジェクトの動きと保存された動きデータとを比較することで、キャラクタ絵を切り替える。

4 まとめと今後の展望

本稿では、Tablescape Animation におけるビジュアルプログラミングを用いたキャラクタ制作ソフトウェアの改良について報告した。なお、今回実装した描画・操作ソフトウェアは、2009 年 11 月 23, 24 日に六本木アカデミーヒルズ 40 にて体験展示を行った。体験者からは、卓上オブジェクトの動きを記録出来ることは、ユーザの希望通りの作品を制作出来るとして評価する声が多かった。その一方、テーブル上での操作とモニタでの設定操作の切り替えの手間が多い、という意見もあった。

このような反応を受けて、今後よりユーザが直感的にアニメーションを制作するために、テーブル上での描画やルール設定、操作を行える環境の実装を検討する予定である。具体的には、デジタルペン等を用いて卓上オブジェクトに直接キャラクタを描画でき、さらに人形型オブジェクト等を使用して、より直感的にルール付けを行える環境作りを考えている。

参考文献

- 1 Y. Kakehi, et al.: "Tablescape animation: a support system for making animations using tabletop physical objects," SIGGRAPH 2009, Talks, Article No. 70, 2009.
- 2 T. Igarashi, et al., "As-Rigid-As-Possible Shape Manipulation," ACM Transactions on Computer Graphics, Vol.24, No.3, ACM SIGGRAPH 2005, pp. 1134-1141, 2005.
- 3 江渡ほか:"Modulobe: 物理シミュレーションによる仮想生物構築環境", EC2005 予稿集, pp. 82-88, 2005.
- 4 Connelly Barnes, et al.: "Video Puppetry: A Performative Interface for Cutout Animation". ACM Transactions on Graphics (Proc. SIGGRAPH ASIA). 27(5), 2008.
- 5 原田ほか: "Viscuit: 柔らかい書き換えによるアニメーション記述言語", 情報処理学会シンポジウム論文集, vol. 5, pp. 183-184, 2004.
- 6 Maloney, J., et al.: "Scratch: A Sneak Preview," Second International Conference on Creating, Connecting, and Collaborating through Computing, pp. 104-109, 2004.

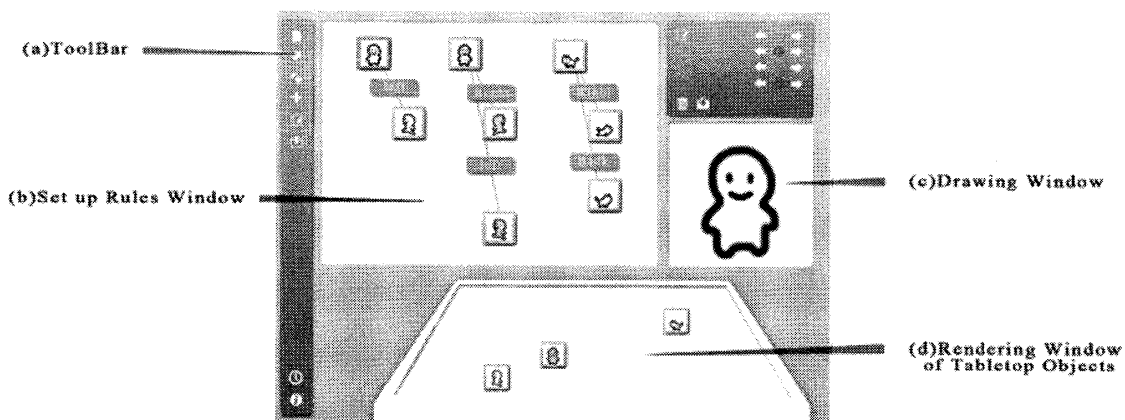


図 2: キャラクタ生成ソフトウェア画面