

5Y-5

# スクリプトによる 3D CG アニメーション生成のための コンポーネントベースのプレイヤーの実装\*

義山琢也<sup>1</sup> 岡田義広<sup>2,3</sup> 新島耕一<sup>2</sup>

<sup>1</sup>九州大学大学院 システム情報科学府

<sup>2</sup>九州大学大学院 システム情報科学研究所

<sup>3</sup>独立行政法人 科学技術振興機構 さきがけ研究「協調と制御」領域

## 1. はじめに

近年、コンピュータハードウェア技術の進歩により 3 次元 CG を実時間で生成可能となり、3 次元 CG アニメーションの需要は、以前よりもさらに高まってきている。しかし、3 次元 CG アニメーションの生成は未だ簡単ではなく、多くの時間を要する作業である。そこで、3 次元 CG アニメーションを自動生成するためのスクリプト言語とそのプレイヤーがいくつか提案されている。そのひとつに TVML(TV program Making Language)[1]がある。

著者らは、3D グラフィックスソフトウェア開発用コンポーネントウェアである *IntelligentBox*[2]を用いて、TVML のプレイヤーを実装している。本稿では、開発しているプレイヤーの機能を説明するとともに、*IntelligentBox* のようなコンポーネントウェアを用いることにより、一から作るよりも小さいコストでプレイヤーの開発が可能であることを述べる。

## 2. プレイヤーの実装方針

*IntelligentBox* における TVML プレイヤーを以下の 3 つの方針に従って実装している。

(1) プレイヤーを *IntelligentBox* の 1 コンポーネントとして実装

プレイヤーをコンポーネントとして提供することにより、複数のスクリプトベースのアニメーションを同時に実行可能となる。TVML のスクリプトの例を図 1 に示す。また、実装したプレイヤーコンポー

```
camera:position(name=A, x=3.0, y=3.0, z=1.0)
character:walk(name=BOB, wait=no)
character:walk(name=TOM)
character:walk(name=KEN, wait=no)
character:talk(name=BOB, text="I'm BOB.")
camera:move(name=A, x=1.0, y=3.0)
```

図 1. TVML スクリプト



図 2. *IBSPlayerBox* の画面スナップショット

ネント *IBSPlayerBox* で、このスクリプトを実行中の画面スナップショットを図 2 に示す。

(2) 言語仕様は、TVML 上位互換

ユーザが対話的な操作を可能のように TVML の言語仕様を拡張する。

(3) *IntelligentBox* の基本部品や合成部品をスクリプトにより制御可能

*IntelligentBox* にはすでに多くの基本コンポーネントやそれらを組み合わせて作ったボックスと呼ばれる部品が用意されている。それらもスクリプトで制御できるように言語仕様を拡張する。

## 3. *IBSPlayerBox* の実装

図 3 に、実装したプレイヤーコンポーネント *IBSPlayerBox* のデータ構造を示す。通常の *IntelligentBox* の部品と同じくスロットに値を保持することで動作する。

ScriptName スロットが変更されると *OpenFileInternal()* メソッドによりスクリプトがファイルから読み込まれ内部に保持される。Play スロット値が True のとき、*IntelligentBox* の TRIGGER メッセージに従って *ExecuteCommand()* メソッドが呼び出され、スクリプトの各行が解釈され実行される。

プレイヤーは、スクリプトに記述された各要素、キャラクタやカメラ・ライトなどを制御する必要があるが、それらはそれぞれ専用のマネージャクラス

\*Implementation of Component Based Player for Script-based 3D CG Animation

Takuya Yoshiyama<sup>1</sup>, Yoshihiro Okada<sup>1,2</sup>, Koichi Nijima<sup>1</sup>

<sup>1</sup>Graduate School of Information Science and Electrical Engineering, Kyushu University

<sup>2</sup>Intelligent Cooperation and Control, PRESTO, JST

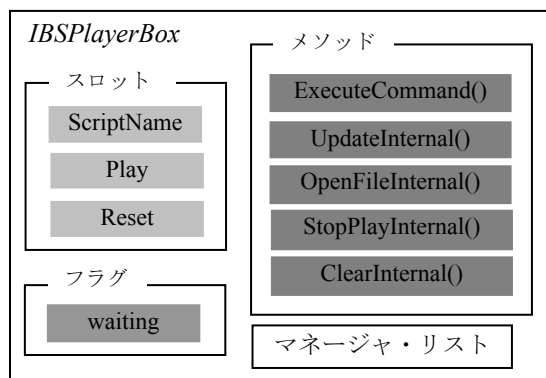


図 3. IBSPlayerBox のデータ構造

を用意することで対処している。つまり、プレイヤー本体の `ExecuteCommand()` メソッドではスクリプトの解釈のみを行い、実際の処理はコマンドの種類によって対応するマネージャが行う。マネージャは、コマンドを受け取るとその内容に従って管理するオブジェクトを操作する。また、コマンドの種類によっては同時に複数のオブジェクトを操作することもあるが、その管理は `waiting` フラグにより行われる。複数のオブジェクトの同時実行は、`IBSPlayerBox` 内で、各マネージャを時分割動作させることにより実現している。時分割動作させる単位として `UpdateInternal()` メソッドが用意されており、定期的呼び出される。この中で、`waiting` フラグに従って、各マネージャの状態更新を行うことにより時分割動作を実現する。

#### 4. オブジェクトの実装

マネージャが管理する様々なオブジェクト、キャラクターやカメラ・ライトなどは `IntelligentBox` の既存コンポーネントを組み合わせることにより実現している。これにより実装のコストを大幅に削減している。カメラオブジェクトの実装を例に説明する。

カメラオブジェクトは、スクリプト中でのシーンをカメラで撮影しているような画面を表示しなければいけないが、これは、`IntelligentBox` に用意されているコンポーネント `CameraBox` を利用することで実現できる。しかし、スクリプトのなかには、カメラの画面にキャラクターのセリフや画面タイトルといった文字列を表示する必要のあるものがある。これは、`CameraBox` のみでは実現できない。そこで、`ClearBoardBox` というコンポーネントを組み合わせることで実現した。`ClearBoardBox` はオブジェクトの表面にテクスチャとして文字列を表示する機能を持ったコンポーネントである。その様子を図 4 および図 5 に示す。`CameraBox` のカメラ画面に一部が重なるように `ClearBoardBox` を配置することにより、カメラ画面にテロップを表示することができる。

他のオブジェクトもできる限り既存コンポーネントを利用することで実現している。ライトオブジ

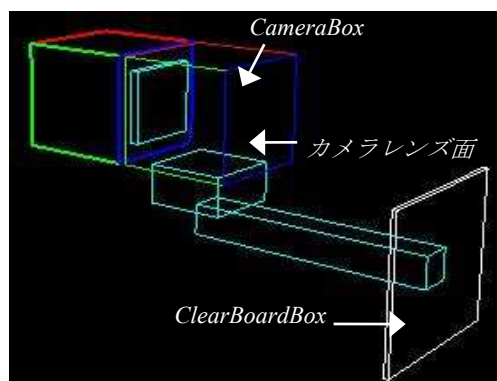


図 4. カメラオブジェクト

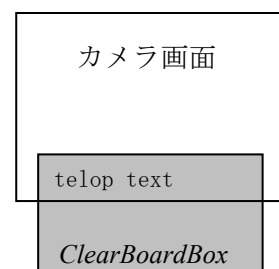


図 5. 文字列表示

ェクトは `LightBox` コンポーネントを利用している。キャラクターオブジェクトは、モーションファイルを読み込む `MotionBox` コンポーネントと `MSRotate3DBox` コンポーネントを各関節にもつキャラクターを組み合わせることでキャラクターのアニメーションを実現する。このように既存コンポーネントを最大限に利用することにより、開発コストを大幅に削減することができた。

#### 5. おわりに

本稿では、TVML プレイヤーの `IntelligentBox` のコンポーネントとしての実装について述べた。本実装では、TVML プレイヤーの機能を `IntelligentBox` の提供するいくつかの既存コンポーネントを組み合わせることで実現している。そのため、スクリプトプレイヤーを一から実装するよりも低コストで開発することができた。今後は、スクリプトの再生のみでなく、対話操作を記録してスクリプトを生成する機能の実装や、TVML 以外のスクリプトプレイヤーの実装を行うつもりである。

#### 参考文献

- [1] Hayashi, M., Ueda, H., Kurihara, T. and Yasumura, M., "TVML (TV program Making Language)-Automatic TV Program Generation from Text-based Script-", Proc. of Imagina'99, 1999.
- [2] 岡田義広, 田中譲: 対話型 3D ソフトウェア構築システム-IntelligentBox-, 日本ソフトウェア科学会コンピュータソフトウェア, Vol.12, No.4, pp.84-94, 1995.