

ヒューマンインタフェースの役割を果たす Java3D による擬人化エージェントの開発に関する検討

吉田 宗史 加藤 誠巳

(上智大学理工学部)

1. まえがき

近年、擬人化されたエージェントがさまざまな分野で利用されている。コンピュータが普及した現在、多機能、高性能化するソフトウェアやサービスに人間が対応しきれないことが考えられる。コンピュータとのインタラクションやコミュニケーションを実現するために、今後エージェントを用いたユーザインタフェースは重要な役割を果たすことになる。

本稿では Java3D を用いたエージェントの開発に関して検討した結果について述べている。エージェントは他のアプリケーションで作成された Java3D 空間にインポートされ、ユーザに対し共通のヒューマンインタフェースを提供することを目的として作成されている。

2. エージェントの機能

筆者らは、音声認識・音声合成を用いた 3DCG 会話ロボット“茶飲み友達”^[1]に関する研究を行っている。システムの開発にあたり、モデルやアニメーションの作成における負担を軽減するために、擬人化エージェントのパッケージ化を行った。ここではエージェントの機能のうち、いくつかを紹介する。

2.1 アニメータクラス

アニメーションを行うために KBKeyFrame、KBRotPosScaleSplinePathInterpolator を用いている。代表となるキーフレームをいくつか設定し、その間を補間させることで実現する。

アニメーションを制御するために Animator クラスを作成した。このクラスによってアニメーションの切り替えや合成、アニメーションデータの再利用が可能となる。シーンの作成の際、開発者は Animator クラスに「歩く」、「止まる」、「座る」などの動作を指示するだけでよい。また、「腕を組みながら歩く」などの合成されたアニメーションも可能となる。

2.2 XML によるアニメーションの記述

アニメーションデータを記述するために XML を用いている。その理由として、必要な箇所だけの動きを記述するだけで済むので作成、編集がしやすい、DOM(Document Object Model) パーサを用いており管理が容易で、目的の動作をすばやく読み込めるなどの利点がある。図 1 にアニメーションの記述例を示す。

```
<?xml version="1.0" encoding="Shift_JIS"?>
<Animator-Data>
.
.
.
<Motion id="50" name="walking" action="elbow_left" length="4">
<keyframe id="0">
<field name="knot" value="0.0"/>
<field name="pitch" value="0.25"/>
</keyframe>
<keyframe id="1">
<field name="knot" value="0.33"/>
<field name="pitch" value="0.40"/>
</keyframe>
<keyframe id="2">
<field name="knot" value="0.66"/>
<field name="position" value="0.0,0.2,0.0"/>
<field name="pitch" value="0.05"/>
</keyframe>
<keyframe id="3">
<field name="knot" value="1.0"/>
<field name="pitch" value="0.25"/>
</keyframe>
</Motion>
.
.
</Animator-Data >
```

図 1 アニメーションの記述例

2.3 視点方向注視

ヒューマンインタフェースとして、エージェントがユーザの方向を振り返るなどの動作は重要な要素のひとつである。Animator クラスには視点方向を注視するためのメソッドが実装されている(lookAtViewPoint)。視点注視にはビルボードを利用している。顔を向けるまでの手順は以下のとおりである(図 3)。

- ①Animator がビルボードと顔の TransformGroup から Transform3D を取り出し、回転コンポーネントを取得
- ②回転コンポーネントから回転角を計算し、キーフレームを作成
- ③作成されたキーフレームを上半身、顔のインタポレータに適用

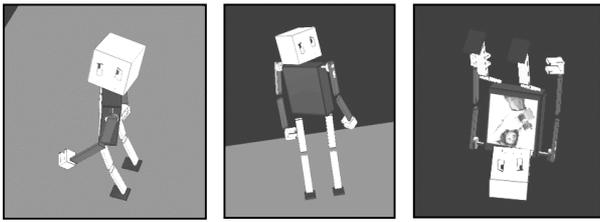


図2 視点注視メソッド実行例

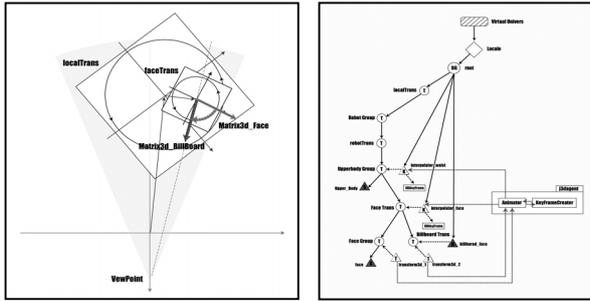


図3 回転角の取得

2.4 スピーカ class

エージェントパッケージには音声同期して口を動かすためのクラス (BelabelaSpeaker) が用意されている。wav 形式のファイルを読み込み、その振幅の一定微小時間内で平均をとり、音量を求める。音量を元にキーフレームを作成し、あごのインタポレータに適用することでアニメーションを実現している。

現在 BelabelaSpeaker クラスには音声ファイルを再生するモード、プログレッシブバーでレベルをモニタするモード、エージェントのあごを動作させるモードの 3 つのモードが用意されている。今後、音声スペクトルを解析し、母音に応じて口の形を変形させるモードを追加することを検討している。

2.5 スケジューラ class

単純に Animator クラスのメソッドを利用してエージェントを操作しても良いが、実際にはシーンを作成する際はスレッドを作成し、そのスレッド内からメソッドを実行している。パッケージにはスレッドから操作を行うためのクラス (Scheduler) が用意されている。

3. パッケージの利用

擬人化エージェントはアプリケーション内やプレゼンテーションで利用することにより、大きな視覚的効果が期待される。たとえば、歩行者経路案内を行うアプリケーションにしても、矢印が移動するよりも人型のモデルが歩いているほうが、イメージしやすい。図4に「画像に

基づく話題を提供するCG会話ロボット・システム」にエージェントを配置した例を示す。図5にエージェントを使用する際の記述例を示す。



図4 エージェントの利用例

```
import j3dagent.*;
...
TransformGroup trans=new TransformGroup();

// CG ロボットの生成
trans.addChild(robotGroup=new RobotGroup(this, trans));

// アニメータクラスの生成
Animator animator=new Animator(robotGroup);
// アニメータクラスをスタート
animator.start();

// スピーカクラスの生成
BelaBelaSpeaker belabela=new BelaBelaSpeaker(
    robotGroup, animator);

// CG ロボットにマイクイベントリスナを追加
robotGroup.addMikeEventListener(new MikeEventListener(){
    // マイク入力があったときの処理
    public void catchVoice(){
        if(!belabela.isRunning()){
            animator.lookAtViewPoint();
            belabela.setSound("./sound/hello.wav");
            belabela.play();
        }
    }
    public void releaseVoice(){}
});
```

図5 記述例

4. むすび

“茶飲み友達”で利用するためのエージェントの開発を行ってきたが、Java3D を使ったどのアプリケーションでもエージェントを利用することができる。本パッケージに限らず、汎用性・ポータビリティのあるものを開発し、ヒューマンインタフェースに利用することを考えている。最後に、有益な御討論をいただいた本学 e-LAB/マルチメディア・ラボの諸氏に謝意を表す。

参考文献

[1] 加藤、谷、桐越、菅田：“音声認識・音声合成を用いた 3DCG 会話ロボット‘茶飲み友達’に関する検討,” 情処第 57 回全大, 6C-03(1998 年 10 月).