

RESTRICCIONES DE ANIMACIÓN ESPECIALIZADAS PARA AMBIENTES 3D

Sánchez, A. – Orosco, R.

UNCPBA – INTIA – DCyS

Campus Universitario – Pje. Arroyo Seco

7000 – Tandil – Argentina

email: {asanchez,orosco}@exa.unicen.edu.ar

1. Introducción

Es reconocida la utilidad de las animaciones en dominios tan diversos como educación, entretenimiento, sistemas de control, sistemas de visualización de información, experimentación científica y otros. Estas facilitan al usuario la comprensión de datos con comportamiento dinámico [I] o la navegación de los mismos. A pesar de esto, los métodos existentes para describir o especificar animaciones aún no se adaptan a las necesidades de programadores no especializados en el área de programación 3D.

El nivel de abstracción provisto por la mayor parte de los toolkits actuales [II] [III] [IV] es una de las principales razones de esta complejidad. Este problema ha sido solucionado parcialmente con técnicas innovadoras tales como las restricciones de animación de Amulet [V], las cuales poseen un estilo declarativo para describir animaciones. Sin embargo, ésta técnica ha sido aplicada mayormente en interacciones gráficas 2D sin tomar en cuenta características especiales y requerimientos de aplicaciones de visualización de información 3D.

En el presente artículo se describe una técnica para la especificación de animaciones. Ésta tiene en cuenta la aplicabilidad de la misma en sistemas de visualización de información y características especiales que surgen en ambientes 3D, tales como la oclusión y la detección de colisiones.

2. Animadores

Un Animador o Restricción de Animación encapsula todos los detalles de implementación de una animación. Por medio de su uso la tarea de especificar una animación se reduce a vincular al atributo del objeto gráfico a animar con el Animador que corresponda al mismo. Por ejemplo, si se desea animar un objeto por medio de una rotación, solo se requiere vincular un Animador con el atributo “ángulo” de tal objeto gráfico. Esto es,

```
// creación de un objeto gráfico
Cube c = new Cube();
// creación de un animador, seteando su valor inicial
Animator anim = new Animator(10);
// vinculo el animador al atributo ángulo del objeto gráfico
anim.attachTo(c, c.getAngle());
```

```
Luego cuando un nuevo valor es asignado al atributo ángulo, con una instrucción como la siguiente,
c.setAngle(90);
```

el animador produce una transición gradual entre el viejo valor y el nuevo, pudiéndose observar la misma como una animación.

De esta forma, incorporar comportamiento dinámico en una presentación no resulta difícil, esto es debido a que el nivel de detalle de la especificación se adecua a las necesidades del programador de la animación.

Como se vio en el ejemplo previo, la especificación de una animación y las acciones que disparan la misma son independientes del tipo de objeto gráfico a ser animado, lo cual permite reutilizar al animador en diferentes aplicaciones.

Los Animadores proveen un conjunto de parámetros con el fin de disponer de un mayor control sobre el comportamiento de los mismos y haciéndolos adaptables a las necesidades de una aplicación en particular. La manera más simple de adaptar un Animador es indicar un componente que especifique la velocidad de animación o su duración (“pacing function”) y otro componente en el cual se puede indicar la función con la cual se calcula la transición entre los valores inicial y final (“path function”). En ambos casos el programador cuenta

con una biblioteca de posibles componentes predefinidos o puede indicar otros distintos con el fin de implementar efectos especiales en la animación.

Los Animadores se clasifican según el tipo de atributo sobre el que trabajan, en el ejemplo previo se trató de un Animador Numérico. También existen Animadores que trabajan sobre listas de valores, colores y listas de puntos.

3. Animadores especializados para ambientes 3D: Animadores Restringidos, Animadores Condicionados y Caminos Alternativos.

Una Restricción Convencional puede ser utilizada para expresar de forma clara y con estilo declarativo relaciones entre los componentes de una escena, tales como posición, colores, luces o cámara. Es posible combinar Animadores con Restricciones Convencionales en un único componente denominado Animador Restringido. Cuando una Restricción Convencional es activada por un cambio en una de las variables envueltas, las otras variables son actualizadas utilizando una animación.

Por ejemplo, es posible mantener a un objeto gráfico durante una presentación a una distancia máxima de la cámara. Una vez introducida la restricción que especifica lo mencionado, si el objeto se aleja a una distancia mayor a la establecida, la cámara modificará su posición pudiéndose observar esto a través de una animación.

A pesar de que el concepto de Animador no depende de si la presentación es 2D o 3D, existen algunas características en presentaciones 3D que deben ser consideradas para desarrollar animaciones que resulten poderosas y efectivas. Algunas son también aplicables a presentaciones 2D pero su presencia en las 3D es de importancia relevante.

Una característica importante en las visualizaciones de información 3D es la oclusión, esto es debido a que si no es tenida en cuenta se pueden producir inesperadamente efectos no deseados. Por ejemplo, un objeto animado en su trayectoria puede ocultar otros objetos de interés durante una animación, perdiéndose el usuario la posibilidad de observar esta información oculta en forma inesperada. Para facilitar la solución a este inconveniente se provee al programador con la posibilidad de especificar para el Animador, en su “path function”, caminos alternativos a seguir durante la animación en el caso de que se produjese la oclusión entre objetos.

Sin embargo la condición de oclusión no siempre se puede prever durante la especificación de la animación. Esta puede depender de atributos en otros elementos de la escena, como pueden ser la posición de los objetos, el punto de vista o el campo de visión. Por lo tanto, disponer de caminos alternativos no es suficiente.

Para estos casos se introduce un nuevo tipo de Animador, el Animador Condicionado, este permite la detección de tales condiciones en tiempo de ejecución. Esto significa que si en medio de una animación la condición del Animador se vuelve verdadera, un método especial del mismo es invocado. Este método puede ser utilizado por el programador para indicar la acción a tomar dada la situación que se ha presentado.

Por medio de los Animadores Condicionados, varias situaciones pueden ser manejadas, las mas comunes son: detección de colisiones, detección de proximidad y la entrada de determinados objetos en un volumen del espacio.

4. Lenguaje de Implementación

Como lenguaje implementatorio se esta utilizando Java junto con el API para gráficos 3D Java3D [VI]. La decisión esta fundamentada en la posibilidad de independizar el trabajo de la plataforma a utilizar, la variedad de posibilidades que brinda para aplicaciones distribuidas, la fácil integrabilidad con otros componentes implementados en el mismo lenguaje, y el constante mejoramiento del hardware para máquinas virtuales y aplicaciones gráficas que nos darán el poder computacional necesario. Como resolovedor de restricciones se esta utilizando Cassowary [VII].

5. Conclusiones

En el presente trabajo se describe el modelo de Animadores y sus especializaciones, el cual pone énfasis en la especificación de animaciones en ambientes 3D y en la aplicabilidad del modelo a software de visualización de información.

Entre las principales ventajas obtenidas en la utilización de Animadores y de Restricciones Convencionales se encuentra el nivel de abstracción de estos que resulta apropiado para el programador no experto, ya que le permite abstraerse de los detalles implementatorios pudiendo concentrarse en el dominio específico de la aplicación en desarrollo. También encontramos la posible reutilización de una previa especificación de animaciones, pudiéndose definir bibliotecas de animadores y sus componentes. En este enfoque se consideran además circunstancias que resultan relevantes en visualizaciones 3D para las cuales se proveen especializaciones de los Animadores.

El objetivo final del trabajo es obtener un modelo genérico para la especificación de animaciones en visualizaciones de información 3D, que resulte adecuado para las necesidades del programador. El trabajo se encuentra en una etapa experimental, trabajando con prototipos de aplicaciones de visualización.

6. Referencias

- [I]. “Information Visualization: Using Vision to Think”. Card, S. Mackinlay, J. Schneiderman, B. Morgan Kauffman, 1999.
- [II]. “OpenGL Programming Guide”. J. Neider, T. Davis, M. Woo, Addison-Wesley, 1993.
- [III]. “The Inventor Mentor: Programming Object Oriented 3D Graphics with Open Inventor”. Wernecke J. Addison-Wesley, 1995.
- [IV]. “Graphics Programming with Direct3D: Techniques and Concepts”. Glidden, R. Addison-Wesley, 1996.
- [V]. “Easily Adding Animations to Interfaces Using Constraints”. Myers, B. Miller, R. McDaniel, R. Ferreny, A. Proc. UIST 96, pp. 119-128, 1996.
- [VI]. “Java3D Specification”. Deering, M. Sowizral, H. Sun Microsystems, 08/1997.
<http://java.sun.com/products/java-media/3D/>
- [VII]. “The Cassowary Linear Arithmetic Constraint Solving Algorithm: Interface and Implementation”. Badros, G. Borning, A. Washington Univ.