ANIMO: Una herramienta para la animación de algoritmos

HERNANDO BUSTOS
Ing. Informático

TEDDY DELGADO
Ing. Informático

MARCO VILLALOBOS
Ing. Informático

e-mail: m_villalobos@utapc2.dnet.uta.cl
e-mail: h_bustos@utapc2.dnet.uta.cl

Depto. de Computación
Facultad de Ingeniería
Universidad de Tarapacá
18 Septiembre # 2222
ARICA - CHILE

Resumen

El presente trabajo contiene información acerca del desarrollo de una herramienta para la animación de algoritmos.

La configuración del proyecto incluye herramientas necesarias para la generación de animaciones, una interfaz gráfica del usuario y controles necesarios para conducir la animación.

El proyecto contempla la interacción con tres componentes básicos; determinación de posiciones claves por animar, generación de escenas de animación, y control de la película. Estos componentes finalmente facilitarán la generación y posterior control de animaciones.

Para el desarrollo del proyecto se ha seguido una metodología Orientada a Objetos (Coad-Yourdon)[COAD91]. El ambiente de trabajo seleccionado corresponde a Microsoft Windows, por ofrecer un ambiente amigable y fácil de usar. Como plataforma de software en la implementación se utiliza Borland C++.

En estos momentos el proyecto se encuentra en fase de desarrollo. El siguiente proyecto se clasifica como COMUNICACION del tipo B.

1. Ambito

El proyecto se enmarca en la optimización, evaluación, enseñanza y desarrollo de algoritmos vía animación por computador.

La animación es una herramienta que puede emplearse en computación gráfica, visualización de algoritmos y tópicos que involucren la misma enseñanza de las ciencias de la computación.

La animación de algoritmo por computador es una herramienta poderosa, ya que puede representar características de un algoritmo en forma más agradable y educativa.

2. Estudio

Las técnicas tradicionales usadas en la comprensión del comportamiento de un algoritmo son las siguientes:

i) Representación textual, en donde se intenta comprender la funcionlidad del algoritmo utilizando su forma textual.

ii) Añadir instrucciones destinadas a generar mensajes que indiquen el estado actual de la ejecución.

iii) Ruteo en papel, en donde se intenta comprender el comportamiento dinámico del algoritmo, simulando la ejecución de instrucciones.

Como alternativa se propone la animación vía computador, cuyas ventajas son: es dinámica, los cambios son efectivamente más rápidos; y la presentación inmejorable.

En este proyecto la animación considera un conjunto de imágenes básicas que serán transformadas, y éstas, en definitiva serán las que definirán la animación (a diferencia de la animación por composición de una series de dibujos estáticos que se trasponen.)

2.1. Técnicas de animación de algoritmos

Las técnicas utilizadas en este proyecto se agrupan en 3 conjuntos que son: técnicas gráficas, de color y audio[BROW85].

Las técnicas gráficas son la base para el proyecto. Las técnicas de color apoyan y complementan a las técnicas gráficas pues añaden otra dimensión para la transmisión de información. Las técnicas de audio son las más recientes y difíciles de utilizar, ellas hacen uso del canal de entrada del audio, mientras que las técnicas gráficas y de color compiten por el canal visual.

Las técnicas gráficas utilizadas son: múltiples vistas, señal de estado, recorrido, transiciones continuas y oportunas, múltiples algoritmos y selección de los datos de entrada.

Las técnicas de color son: codificar el estado de las estructuras de datos, destacar la actividad, unificar múltiples vistas, enfatizar patrones y hacer visible el recorrido.

Las técnicas de audio son: reforzamiento visual, expresar patrones, sustituir visualizaciones y señalar condición de excepción.

3. **Propuesta**

La animación de un algoritmo es el proceso de abstracción de los datos, operaciones y semántica del programa, y la creación dinámica de vistas gráficas de aquellas abstracciones.

Para lograr una animación efectiva, es necesario un proceso establecido para lograr uniformidad en el desarrollo de las aplicaciones. Los siguientes pasos resumen lo anterior:

* definir operaciones en un programa.
* diseñar acciones de animación para simular esas operaciones,
* organizar y controlar la realización de la animación.

El proyecto contempla una estructura con tres componentes. Debido a que la animación de algoritmos es una visualización dinámica de las operaciones y datos de un programa, la estructura contiene un componente de algoritmo y un componente de animación. Y por último, se necesita controlar en alguna medida la realización misma de la animación, la cual se persigue con un componente de ejecución.

Se consideró y modificó el ***paradigma transición-camino***[STAS90] para el diseño del proyecto. Este paradigma está ahora basado en seis clase-objetos: posición, ventana, imagen, camino, transición y asociación. Las clase-objetos y sus servicios proveen una interfase general de diseño de animación sin considerar el programa a ser animado o el hardware sobre el cual operará la animación. Aunque se tengan que escribir programas (pues se debe describir el comportamiento de la animación por cada componente de algoritmo), ya no se tendrá que hacer llamadas gráficas de bajo nivel y dependientes del sistema.

3.1. **Componente de algoritmo**

Para dirigir o activar una animación, se identifican las posiciones claves en el programa, correspondientes a aquellas operaciones que son importantes para la semántica del programa, tal como las operaciones de comparación e intercambio en un algoritmo de ordenamiento. Tales posiciones se llamarán componentes de algoritmo, y son aquellas operaciones que involucran una acción de importancia (tal vez no evidente) dentro del algoritmo. El componente de algoritmo es una *conceptualización* que será modelada por el componente de animación.

3.2. **Componente de animación**

El componente de animación contiene los objetos gráficos que cambian completamente de posición, tamaño, color, etc. entre marcos de una animación. Y las operaciones que gobiernan los cambios para controlar una acción. Este modelo formal del componente contiene seis clase-objetos: imágenes gráficas, posiciones de las mismas, ventanas por donde serán desplegadas, transiciones, caminos y asociaciones.

Los caminos y transiciones modifican los atributos de posiciones e imágenes. Las asociaciones proveen un mecanismo para almacenar consultar y recuperar imágenes y ventanas. Se produce una animación creando, manipulando y editando instancias de las clase-objetos y se incluye un conjunto de servicios para definir y modificar sus instancias. La Tabla N°1 muestra los servicios por cada clase-objeto.

A continuación se describirá en una forma más detallada cada uno de las clase-objetos definidas:

i) **Imagen**

Objeto gráfico que experimenta cambios en su posición, tamaño, etc., a través de los

cuadros de una animación. Como imágenes básicas se tienen líneas, rectángulos, círculos, y texto.

ii) **Posición**

Se define una posición como un par coordenado real (x, y).

iii) **Ventana**

Espacio habilitado para mostrar y controlar una animación.

iv) **Camino**

Designa la magnitud del cambio en los atributos de una imagen. Se define como una secuencia finita de pares coordenados (x, y) cuyos valores son todos positivos.

v) **Transición**

Las transiciones definen la forma de modificación que recibirán las imágenes a través de caminos, y son las que describen las escenas de animación. Las transiciones básicas reciben como parámetro un tipo de transición, la ventana relacionada, una imagen y un camino que especifica la modificación. Las transiciones básicas son: movimiento, escalamiento, rellenado, color.

Existen también transiciones complejas. El servicio Composición permite ejecutar transiciones concurrentemente, alternando marcos de animación de dos transiciones. El servicio Iterar copia una transición en sí misma ***n*** veces. El servicio Concatenar añade una transición a otra. Y el servicio Realizar "ejecuta" el resultado final.

vi) **Asociación**

Las asociaciones proveen un mecanismo general para almacenar, consular y recuperar instancias de objetos. La estructura es de uso general y aceptará cualquier tipo de referencia.

3.3. Componente de ejecución

Obtenidas las escenas de animación, éstas son agrupadas en una estructura llamada ***película***, la cual permite al usuario controlar la ejecución de la animación. Este componente es el que finalmente mostrara los resultados y permitirá concluir a partir de una animación. Dos son los pasos involucrados en este componente; generación de la película y control de la misma.

i) **Generación de la película**

El servicio Realizar (dentro del componente de animación) funciona como un verdadero filtro (pues interrelaciona los componentes de animación y de ejecución), es el encargado de añadir las escenas de animación a la película.

La estructura de la película es casi idéntica al de las transiciones, en donde los atributos X e Y son manejados dependiendo del tipo de transición, esto para habilitar un recorrido bidireccional a través de los marcos de la película.

ii) **Control de la película**

Los controles son los que utilizará el usuario para manejar y conducir la ejecución de la animación. Estos incluyen la posibilidad de redefinir el punto de partida y llegada de la animación (puntos lógicos), movimientos básicos a través de la película (al inicio, al final, un marco atrás o uno adelante) y aquellos que permiten iniciar y finalizar una animación.

Una vez generada la película se tiene una serie de controles que permitan al usuario manejar la animación, los que son:

a)	**Play:**	activa la animación desde el punto actual.
b)	**Stop:**	detiene la animación.
c)	**Forward:**	avanza un marco.
d)	**Rewind:**	retrocede un marco.
e)	**SpeedUp:**	acelera la velocidad de animación.
f)	**SpeedDown:**	disminuye la velocidad de animación.
g)	**First:**	se dirige al primer marco lógico de animación.
h)	**Last:**	se dirige al último marco lógico de animación.
i)	**Top:**	redefine el inicio lógico de la película.
j)	**Botton:**	redefine el final lógico de la película.
k)	**Exit:**	finalizar la ejecución.

Posición	Ventana	Imagen	Camino	Transición	Asociación
Crear Ubicar X Y Borrar	Crear X Y Ancho Alto Plano Fondo	Crear Ubicar X Y	Crear Largo Ejemplo Nulo Copiar X Y Borrar	Crear Iterar Concatenar Componer Realizar Borrar	Agregar Eliminar Recuperar Existe

Tabla N°1 Servicios por cada clase-objeto

4. Conclusiones

- Este proyecto soportará la animación por computador y complementará las funciones tanto de programadores, animadores y usuarios en las tareas de enseñanza, optimización, evaluación y desarrollo de algoritmos.

- De acuerdo al desarrollo actual del proyecto, se está cumpliendo con uno de sus objetivos principales el cual es, justamente, la enseñanza de las mismas ciencias de la computación, ya que se entrega un método radical pero eficiente, efectivo, didáctico, cómodo y fácil de utilizar.

- La estructura de la herramienta es tal, que efectivamente permite agregar fácilmente nuevas imágenes básicas y tiene la capacidad de soportar modificaciones sin mayores incidencias en otros componentes, dado el uso de tecnologías avanzadas tales como método de diseño con orientación a objetos, uso de lenguajes C++ y técnicas de animación basadas en experiencia en otros sistemas de similar aplicación.

5. Referencias

[BROW85] M. Brown y R. Sedgewick. "Techniques for Algorithm Animation". IEEE Software, Vol. 2 No. 1 Jan. 1985, pp. 28-39.

[STAS90] John Stasko. "Tango: A Framework and System for Algorithm Animation". Computer, Vol 23 No. 9 Sept. 1990, pp. 27-39.

[ROMA93] Gruia-Catalin Roman and Kenneth C. Cox. "A Taxonomy of Program Visualization Systems". Dec. 93, pp. 11-24.

[COAD91] Peter Coad and Edward Yourdon. "Objecto-Oriented Analysis". Prentice-Hall International, Inc., 1991.