

Biblioteca de Efectos de Transición de Pantallas para Aplicaciones de Realidad Virtual

Autor: Yoander Cabrera Díaz

Universidad de las Ciencias Informáticas, La Habana, Cuba, ycabrerad@uci.cu

Co-Autor: Ronaldo Castro Milán

Universidad de las Ciencias Informáticas, La Habana, Cuba, rcastro@uci.cu

Mentor: MSc. Omar Correa Madrigal

Universidad de las Ciencias Informáticas, La Habana, Cuba, ocorrea@uci.cu

RESUMEN

La realidad virtual (RV) se ha expandido considerablemente a diferentes sectores en estos últimos años, puede encontrarse en aplicaciones militares, la medicina, entretenimiento, ofreciendo servicios de manera interactiva y segura. Esto ha provocado la evolución del hardware, y del software usado en sistemas de visualización, simulación y efectos de transiciones que se pueden realizar con imágenes o planos para dar realismo a este tipo de aplicaciones.

El trabajo se resume en realizar un estudio de las técnicas de transición de pantallas empleadas en el Software Microsoft Power Point, con vista a su futura implantación en aplicaciones de realidad virtual, específicamente en video-juegos que se desarrollan en el Grupo de Proyectos de Juegos Virtuales (GPJV) de la Facultad 5 (F5) de la Universidad de las Ciencias Informáticas (UCI) de Cuba. Los resultados de este estudio serán aplicados en una biblioteca la cual será capaz de aportar a los programadores del GPJV mayor facilidad, comodidad y rapidez a la hora de concebir interfaces visuales que necesiten la inclusión de este tipo de técnicas.

Palabras Clave: Biblioteca, Efectos, Transición, Pantalla, Realidad Virtual.

ABSTRACT

The virtual reality (RV) has expanded considerably in different sectors in recent years, can be found in military applications, medicine, entertainment, offering services interactively and secure. This has led to the evolution of hardware and software used in display systems, simulation and effects of transitions that can be done with pictures or drawings to give realism this type application.

The work is summarized in conducting a study of the techniques employed in transition screens Software Microsoft Power Point, with a view to its eventual installation of application in virtual reality, specifically in video-game's developed by the Virtual Games Project Group, faculty 5 (F5) at University of Informatics Sciences (UCI). The results of this study will be applied in a library which will be capable of providing programmers the GPJV greater ease, comfort and speed in the design visual interfaces required by the inclusion of such techniques.

Keywords: Library, Effects, Transition, Screen, Virtual Reality.

1. INTRODUCCIÓN

Las transiciones de pantallas son efectos visuales que cambian la apariencia gráfica de una aplicación, permiten esa interactividad de cambiar de un estado de vista a otro, lo cual no es más que la forma en que una imagen cambia a otra en una animación o una presentación de una multimedia o en las interfaces de los videos-juegos, la definición de las transiciones depende en medida de las clases de efectos que se utilicen.

Los efectos de transición aunque están muy vinculados a múltiples esferas de la computación tienen una estrecha relación con las aplicaciones de realidad virtual, sus vínculos están muy estrechados, su empleo permite jugar con efectos sorprendentes en cada interfaz, permite gran dinamismo en la apariencia de las aplicaciones y en las confecciones de menús.

En la actualidad la **situación problemática** radica en:

- Los programadores invierten mucho tiempo construyendo interfaces visuales.
- La elaboración de las interfaces no se logra de una forma fácil e intuitiva, además es inevitable en ocasiones

repetir código fuente.

- La modificación de código fuente para realizar mejoras en las aplicaciones, arroja dificultades.
- Las aplicaciones no quedan con niveles de inmersión y enriquecimiento visual aceptable para los usuarios

De lo anteriormente planteado, se desprende como interrogante el siguiente **problema científico** a resolver:

¿Cómo desarrollar una herramienta que permita la elaboración de menú con efectos similares a las transiciones de diapositivas del Power Point para obtener interfaces visuales en tiempos aceptables y con calidad visual?

Como **objetivo general**, se propone desarrollar una biblioteca de transición de pantallas capaz de facilitar el trabajo a los programadores y brindar resultados visuales aceptables para los usuarios.

2. DESARROLLO

2.1. TRANSICIÓN ENTRE IMÁGENES

Permite el cambio de una imagen a otra, añadiendo un efecto visual que se reproduce entre dos fotos para que la transición de imágenes sea más suave, más atractiva o más interesante (iPod ABC, 2007).

2.2. TRANSICIÓN ENTRE PANTALLAS

Permite especificar cómo se va a desarrollar el cambio de una pantalla a otra y suavizar las sucesiones. Mediante esta se puede utilizar el mismo efecto en todas las transiciones o aplicar diferentes modos de unas a otras (Pérez, 2007). Es preciso organizar el paso de una pantalla a la siguiente de una forma variada, pero dentro de una lógica que le de unidad a toda la presentación. La animación dentro de la pantalla permite establecer el orden y la forma en la que van a ir apareciendo los objetos que la constituyen (Bravo, 2002).

2.3. EFECTOS DE TRANSICIÓN

Las transiciones son unos de los detalles que hace comparar una aplicación de buena realización, son pequeños puntos de detalles que ayudan hacer los sistemas de realidad virtual (SRV) más divertidos e interesantes.

2.4. TÉCNICA DE TRANSPARENCIA. ALPHA BLENDING

Alpha Blending es una técnica muy utilizada para lograr las transiciones entre imágenes, es un factor esencial a la hora de desarrollar una transición entre escenarios, permitiendo obtener una gran variedad de efectos. Es el proceso de combinar dos objetos en pantalla teniendo en cuenta los valores alfa que designa su grado de transferencia, posibilitando que el color del píxel del objeto mostrado en segundo plano se vea influenciado por el grado de transparencia del píxel correspondiente al situado en primer plano. Permite crear objetos transparentes de forma que cualquier objeto situado detrás de él sea visible respetando su opacidad (Efectos Gráficos, 2007).

Esta técnica se compone por dos elementos: Alpha componente principalmente utilizado para especificar el nivel de transparencia de un píxel y Blending que permite mezclar los píxeles que estamos dibujando con otros que fueron dibujados previamente. Se asume que se reserva 8 bits para el valor alfa de cada píxel, donde el intervalo válido para el componente alfa sería [0,255] la cual corresponde a una opacidad de [0%,100%], se obtiene la transparencia con valores blanco (255), el píxel sería completamente opaco y la semi-transparencia con valores grises. La idea del Blending es combinar los píxeles, un primer píxel que se está escribiendo llamado píxel de origen (source píxel) con otro píxel de destino (destination píxel). Se deben dibujar los objetos que no utilizan Blending primero. Luego se deben ordenar los objetos que usan Blending por su distancia a la cámara; esto se puede lograr más eficientemente si los objetos están en el espacio de vista (ya transformados para la pantalla), así que se puede simplemente ordenar el componente Z. Finalmente, se deben dibujar los objetos que utilizan Blending de atrás hacia delante (GDA, 2007).

2.5. SOLUCIÓN PROPUESTA

2.5.1. DESCRIPCIÓN DE LA SOLUCIÓN

Dada la necesidad de obtener una biblioteca que maneje los diferentes efectos de transición de pantallas y que esta pueda ser de uso en diferentes plataformas, se hará uso de la API (del inglés *Application Programming Interface*) OpenGL teniendo en cuenta las posibilidades y ventajas que brinda.

La biblioteca usará como estructura de datos básica el Vector de la STL (del inglés *Standard Template Library*) de C++, por las facilidades que brinda dicha estructura. El uso de esta lista de efectos permite una cómoda manipulación y acceso, aprovechando también la gestión dinámica de la memoria que hace, optimizando de esta forma el trabajo con la biblioteca.

El lenguaje de programación que se empleará es el C++ debido a su alto nivel, el cual incorpora una variedad de características que facilitan una programación elegante y modular, además el código escrito en C++ estándar permite mantener su portabilidad hacia otras plataformas.

2.5.2. DESCRIPCIÓN ALGORÍTMICA DE LOS EFECTOS DE TRANSICIÓN SELECCIONADOS.

Cada transición posee un proceso de realización en el cual implica el trabajo con textura, pintar primitiva y ejecutar el efecto de acuerdo a la velocidad con que se quiere realizar el desplazamiento, donde cada uno se realiza de diferentes formas.

Transición Aparecer

Este efecto se realiza entre dos imágenes que son cargadas como texturas, y asignada a dos planos que cada uno tiene la forma de cuadrado, cada plano texturizado tendrá como función de OpenGL, `glBlendFunc (GL_ONE, GL_ONE)` para plano texturizado del frente y la textura del plano de fondo `glBlendFunc (GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)`, para activar los valores que va contener la función Alfa Blending para mezclar los pixeles de origen y destino, y lograr la técnica de transparencia.

Posteriormente se va graduando la opacidad (el valor de alfa que significa Z) de la textura que se quiere que aparezca, en este caso, la textura del plano de fondo. La opacidad se calcula:

```
s= (timeII-timeI)*velocity
opacity +=s
Si opacity mayor 1.0
Entonces:
opacity=1.0
Sino
Si opacity menor 0.01
Entonces:
opacity igual a 0.01
d igual a 0.01
RedibujaVentana
```

Este valor de opacidad se le pasa a la función `glColor4f (1.0, 1.0, 1.0, opacity)` del cuadrado de fondo y se logra el efecto. Ejemplo relacional de las funciones para el cuadrado de fondo:

```
glEnable (GL_BLEND);
glBlendFunc (GL_SRC_ALPHA,
GL_ONE_MINUS_SRC_ALPHA);
glBegin(GL_QUADS);
glNormal3f (0.0f, 0.0f, 1.0f);
glColor4f (1.0, 1.0, 1.0, opacity);
glVertex3f (-vertexX,-vertexY, 0.0);
glVertex3f (vertexX,-vertexY, 0.0);
glVertex3f (vertexX, vertexY, 0.0);
glVertex3f (-vertexX, vertexY, 0.0);
glEnd();
```

Transición Giro desde el Centro

Este efecto de transición se realiza entre dos imágenes que son cargadas como texturas, y asignada a dos planos que cada uno tiene la forma de cuadrado, se inicializan las dimensiones del fondo con valores menores que el frente, para que de forma gradual ir escalando los vértices del plano de fondo hasta que llegue a sobreponerse al plano del frente con igual valor de los vértices, a la vez se va rotando el plano del fondo con respecto al eje Z. Para lograr el efecto de giro, los valores de los vértices, ángulo de rotación, y eje Z se calculan de esta forma:

```

s= (timeII-timeI)*velocity
angle+= (360/s)
increment+=s
z=increment
vertexX= increment
vertexY= increment
Si increment mayor sizeQuad
Entonces:
vertexX igual a sizeQuad
vertexY igual a sizeQuad
angle igual a cero
RedibujaVentana

```

Los valores de los vértices que se calculan y se le pasa a la función `glVertex3f (vertexX, vertexY, 0.0)` a la hora de construir el plano del fondo.

Ejemplo:

```

glBegin (GL_QUADS);
glNormal3f (0.0f, 0.0f, 1.0f);
glVertex3f(-vertexX,-vertexY,0.0);
glVertex3f(vertexX,-vertexY,0.0);
glVertex3f(vertexX, vertexY,0.0);
glVertex3f(-vertexX,vertexY,0.0);
glEnd();

```

Y los valores del eje Z y ángulo de rotación que se le pasa a la función de `glRotatef` (función que permite rotar respecto a un eje y con un ángulo determinado) sería `glRotatef (angle, 0.0, 0.0, z)`, se le indica esta función al plano de fondo antes de construirlo, de esa forma el plano texturizado del fondo se interpone al plano texturizado de frente rotando y aumentando sus vértices. Ejemplo relacional de las dos funciones:

```

glRotatef (angle, 0.0, 0.0, z);
glBegin(GL_QUADS);
glNormal3f (0.0f, 0.0f, 1.0f);
glVertex3f (-vertexX,-vertexY,0.0);
glVertex3f (vertexX,-vertexY, 0.0);
glVertex3f (vertexX, vertexY, 0.0);
glVertex3f(-vertexX, vertexY, 0.0);
glEnd();

```

Transición Girar y Encoger

El sentido de giro y encoger se realiza entre dos planos, a cada uno le corresponde una textura, se define el plano frente siendo el objetivo clave de aplicarle una rotación respecto a un ángulo de rotación y el eje Z, a la vez se disminuye los vértices para que de paso a la aparición del plano que está de fondo. Los valores para lograr la rotación y el encogimiento se calculan como se muestra a continuación:

```

s= (timeII-timeI)*velocity
angle+= (360/s)
increment+=s
z=increment
vertexX= sizeQuad - increment
vertexY= sizeQuad - increment
Si increment mayor sizeQuad
Entonces:
vertexX igual a cero
vertexY igual a cero
angle igual a cero
RedibujaVentana

```

Una vez calculado los valores del eje Z y el ángulo de rotación se le pasa a la función de rotación, actualizando a la misma vez los vértices del plano del frente, logrando de esa forma el efecto de girar y encoger a la vez.

Transición Estirar desde el Centro

Este efecto de transición se realiza entre dos imágenes que son cargadas como texturas, y asignada a dos planos que cada uno tiene la forma de cuadrado, se construye primero el plano del frente y posteriormente se va aumentando las dimensiones de los vértices del plano de fondo en forma de estiramiento, los vértices se van estirando hasta que escale completo la textura del plano de frente y se quede el plano de fondo como frente. Para eso se calculan las dimensiones del plano de fondo como se muestra a continuación:

```
s= (timeII-timeI)*velocity
stretch+=s
vertexX= stretch
vertexY= stretch
Si stretch mayor sizeQuad
Entonces:
stretch igual a sizeQuad
RedibujaVentana
```

Una vez calculado los valores de los vértices del plano de fondo se construye el plano y se carga su textura, logrando que sus vértices aparezcan en forma de estiramiento al actualizarle cada uno de los valores que van ir tomando, hasta lograr interponerse ante el plano texturizado del frente.

Transición Estirar desde la esquina abajo a la izquierda

Se realiza entre dos imágenes que son cargadas como texturas, y asignada a dos planos que cada uno tiene la forma de cuadrado, se construye primero el plano de frente, posteriormente el plano de fondo con una dimensión mucho más pequeña que el plano de frente, con el objetivo de que el plano de fondo vaya estirando sus dimensiones, a la vez que se traslada por los ejes X e Y para poder cubrir el plano texturizado de frente y se quede el plano que viene trasladándose como fondo, para obtener los valores de las dimensiones del plano de fondo se tiene:

```
s= (timeII-timeI)*velocity
stretchDleft+=s
traslDLX=traslDLX - stretchDleft
traslDLY=traslDLY - stretchDleft
vertexX= vertexX + stretchDleft
vertexY= vertexY + stretchDleft
Si traslDLX menor igual 0 y vertexX mayor igual
sizeQuad
Entonces:
traslDLX igual a cero
traslDLY igual a cero
vertexX igual a sizeQuad
vertexY igual a sizeQuad
RedibujaVentana
```

Una vez que se obtienen los valores de traslación y las dimensiones se va repintando la escena del plano de fondo, moviéndose gradualmente y estirándose hasta que logre interponerse al plano texturizado del frente, los valores de la traslación tienen que ser asignado a la función de la traslación como negativos para poder posicionar el plano en la posición de la transición en este caso estiramiento desde abajo a la izquierda.

Transición Estirar desde la Esquina de Abajo a la Derecha

Este efecto de transición se realiza entre dos imágenes que son cargadas como texturas, y asignada a dos planos que cada uno tiene la forma de cuadrado, se construye primero el plano del frente, posteriormente el plano de fondo con una dimensión mucho más pequeña que el plano de frente, con el objetivo de que el plano de fondo vaya estirando sus dimensiones y trasladándose desde la derecha a la izquierda por los bordes del plano de frente hasta interponerse y quedar como fondo, los pasos para obtener los valores de la traslación y dimensión del plano de fondo son:

```

s= (timeII-timeI)*velocity
stretchDrigh+=1/s
trasIDRX=trasIDRX - stretchDrigh
trasIDRY=trasIDRY - stretchDrigh
vertexX= vertexX + stretchDrigh
vertexY= vertexY + stretchDrigh
Si trasIDRX menor igual 0 y vertexX mayor igual
sizeQuad
Entonces:
trasIDRX igual a cero
trasIDRY igual a cero
vertexX igual a sizeQuad
vertexY igual a sizeQuad
RedibujaVentana

```

La obtención de estos valores van a tener la misma idea que el caso anterior de transición solo cambia la filosofía de pintar el plano, cambia el signo del vértice x en este caso sería negativo y el valor del eje y y de traslación también, para poder posicionar el plano de fondo a la derecha y trasladar uniformemente el plano hasta que logre interponerse al otro plano que está en el frente.

Transición Estirar desde la Esquina de Arriba a la Izquierda

La transición se realiza entre dos imágenes que son cargadas como texturas, y asignada a dos planos que cada uno tiene la forma de cuadrado, se construye primero el plano del frente, posteriormente el plano del fondo con una dimensión mucho más pequeña que el plano de frente, con el objetivo de que el plano de fondo vaya estirando sus dimensiones y trasladándose desde arriba de izquierda a derecha.

Los pasos para obtener las dimensiones y los valores de traslación del plano de fondo es el mismo caso de los efectos de la familia de estiramiento desde abajo, el cambio radica en la posición donde debe estar el plano para que comience su transición desde arriba, para eso solo se cambia el sentido del valor de traslación del eje X , se escala como negativo para poder lograr mover el plano desde ese sentido hasta que se interponga al plano del frente.

Transición Estirar desde la Esquina de Arriba a la Derecha

Se efectúa entre dos imágenes que son cargadas como texturas, y asignada a dos planos que cada uno tiene la forma de cuadrado, se construye primero el plano del frente, posteriormente el plano de fondo con una dimensión mucho más pequeña que el plano de frente, con el objetivo de que el plano de fondo vaya estirando sus dimensiones y trasladándose desde arriba de derecha a izquierda. Los pasos para obtener las dimensiones y los valores de traslación del plano de fondo es el mismo caso de los efectos de transición de la familia de estiramiento de abajo, el cambio radica en la posición donde debe estar el plano para que comience su transición desde arriba, para eso solo se cambia el sentido del valor de traslación del eje x e y , se escala como positivo al igual que los vértices del plano, para poder lograr mover el plano desde ese sentido hasta que se interponga al plano de frente.

Transición Deslizar desde la Derecha

La transición tiene lugar entre dos imágenes que son cargadas como texturas, y asignada a dos planos en forma de cuadrado, se construye primero el plano de fondo y se traslada a otra posición que no sea el centro de visión, posteriormente se construye el plano de frente en la posición del centro de visión, la idea de esta transición es: ir deslizando o empujando el plano de fondo desde la derecha hasta que se interponga al plano de frente quedando el plano de fondo como frente, para eso se debe ir trasladando por el eje x positivo, el plano de fondo utilizando la función $glTranslatef(ejeX,ejeY,ejeZ)$, para calcular los valores del eje x se tiene:

```

s= (timeII-timeI)*velocity
Si moveRX mayor que cero
Entonces:
moveRX=moveRX-s
Sino
Si moveRX menor que cero
Entonces:
moveRX=0.0
RedibujaVentana

```

Como resultado de este cálculo se van obteniendo los valores de traslación en el eje x positivo del plano de fondo, hasta que este se interponga al plano de frente que es cuando $movRX$ sea cero, indica que el plano de fondo se encuentra en la posición del centro de visión.

Transición Deslizamiento desde la Izquierda

Este efecto de transición tiene las mismas características que el efecto tratado anteriormente (efecto de transición deslizamiento desde la derecha), su variación se encuentra en la forma que se va a efectuar la traslación del plano de fondo, el valor de traslación es por el eje x y negativo, indicando el sentido en que se va a realizar el empuje o traslación, a la izquierda hasta que llegue a interponerse al plano del frente.

El cálculo de los valores de traslación del eje x es el mismo que en el caso anterior, termina el efecto de transición cuando el valor del eje x ($movRX$) sea cero y de esa forma el plano de fondo pasa a ser el frente de visión.

La función de traslación quedaría de esta forma:

```
glTranslatef(-movRX,ejeY,ejeZ)
glBegin(GL_QUADS);
glNormal3f(0.0f, 0.0f, 1.0f);
glVertex3f(-vertexX,-vertexY, 0.0);
glVertex3f(vertexX,-vertexY, 0.0);
glVertex3f(vertexX,vertexY, 0.0);
glVertex3f(-vertexX, vertexY, 0.0);
glEnd();
```

Transición Deslizamiento desde Arriba

Se realiza entre dos planos en forma de cuadrado, cada uno carga una imagen como textura, inicialmente se construye el plano de fondo, y se traslada en una posición no visible al centro de visión, en este caso con respecto al eje y , y después se construye el plano de frente visible al centro de visión. El deslizamiento se realiza en otro sentido respecto al eje Y positivo desde arriba hacia abajo, de esa forma va apareciendo el plano de fondo empujando en ese sentido hasta que se interponga al plano de frente, la función de acción del empuje es la traslación $glTranslatef(ejeX,movUY,ejeZ)$.

De la siguiente manera se calcula el sentido de empuje:

```
s= (timeII-timeI)*velocity
Si movUY mayor que cero
Entonces:
moveUY= moveUY- s
Sino
Si movUY menor que cero
Entonces:
moveUY=0.0
RedibujaVentana
```

Como resultado de este cálculo se van obteniendo los valores de traslación en el eje y positivo del plano de fondo hasta que se interponga al plano de frente que es cuando $movUY$ sea cero, indica que el plano de fondo se encuentra en la posición del centro de visión.

Transición Deslizamiento desde Abajo

Tiene las mismas características que el (efecto de transición deslizamiento desde arriba), su variación está en la forma en que se va a efectuar el empuje o la traslación, por el eje y negativo indicando el sentido de la traslación desde abajo hacia arriba hasta que se interponga el plano de fondo al plano de frente.

El cálculo de los valores de traslación del eje y negativo es el mismo que en el caso anterior, el efecto de transición termina cuando el valor del eje y ($movUY$) sea cero y de esa forma el plano de fondo pasa a ser el frente de visión. La función de traslación quedaría $glTranslatef(ejeX,-movUY,ejeZ)$.

2.6. MODELO DE DOMINIO

El diagrama que a continuación se muestra representa el modelo de dominio de la biblioteca. Este modelo ayuda a los programadores a la comprensión y análisis de la relación conceptual a diseñar. Véase la figura 1.

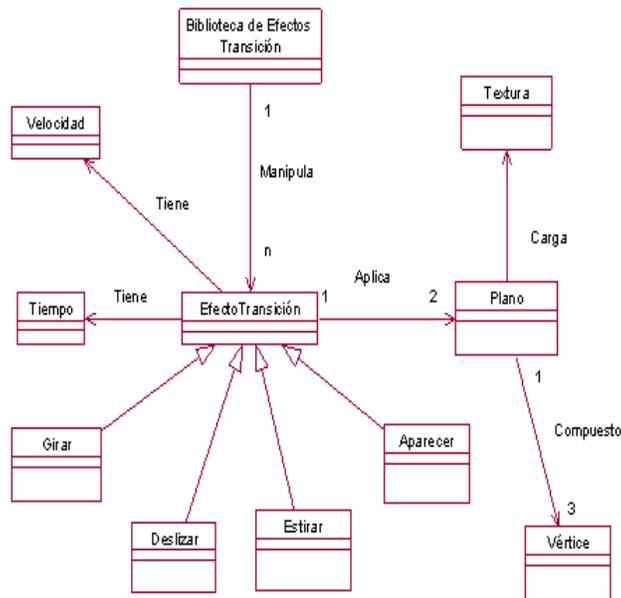


Fig. 1: Modelo de Dominio

2.6.1. DESCRIPCIÓN DE DOMINIO

A continuación se presentan un conjunto de conceptos que conforman la terminología del modelo de dominio, con el propósito de facilitar un mayor entendimiento de los términos manejados en el diagrama anterior.

Biblioteca: Controlador o manipulador de los efectos de transición y sus propiedades.

Velocidad: Rapidez de cambio del efecto que se efectúe y valor determinante para calcular conjuntamente con el tiempo el desplazamiento del efecto de transición.

Tiempo: Tiempo del sistema que se va a tener en cuenta para calcular el desplazamiento que requiere el Efecto de Transición para poder realizarse.

Plano: Figura geométrica de visibilidad de la escena que se le carga una textura para poder aplicar el efecto de transición.

Textura: Imagen de la superficie del plano, que es cargada a cada plano que se crea.

Girar: Efecto de transición que rota un plano determinado y a la vez amplía sus dimensiones hasta poder interponerse al plano que se encuentra como frente de visión.

Deslizar: Efecto de transición que va deslizando un plano determinado encima de otro plano hasta interponerse como frente de visión actual.

Estirar: Efecto de transición que aumenta los vértices del plano determinado hasta interponerse al plano que está como centro de visión.

Aparecer: Efecto de transición que permite la visibilidad de un plano determinado.

Vértice: Valores que se van a tener en cuenta para construir el plano en el eje de coordenada, dimensiones del plano.

2.7. ARQUITECTURA

La arquitectura de la biblioteca, está basada en el patrón arquitectónico N-Capas, se implementan dos capas lógicas fundamentales. La capa de **Presentación** que contiene las clases que manejan la información de los efectos de transición que posee la biblioteca; y la capa de **Lógica de Negocio**, que engloba las clases que implementan cada uno de los efectos de transición. El objetivo de estas clases es recibir y ejecutar las solicitudes que se hagan de la capa de **Presentación** y devolverle a esta un resultado. Véase la figura 2.



Fig. 2: Representación de la arquitectura

2.8. INTEGRACIÓN

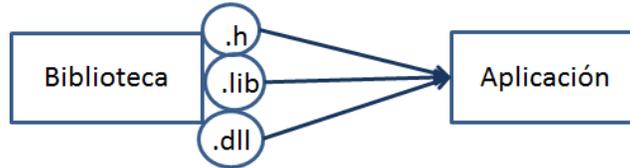


Fig. 3: Integración de la biblioteca con las aplicaciones

3. RESULTADOS



Fig. 4: Transición Aparecer



Fig. 5: Transición Estirar



Fig. 6: Transición Deslizar



Fig. 7: Transición Girar

La biblioteca fue utilizada para desarrollar el menú del video-juego “Rápido y Curioso”, desarrollado en el GPJV. Las figuras anteriores pertenecen a dicho video-juego.

4. CONCLUSIONES

- Se valoraron las técnicas, tecnologías y tendencias en cuanto al uso de los efectos de transición de pantallas lo cual contribuyó a revelar nuevas características para su uso.
- Se realizaron pruebas con programadores para verificar si la biblioteca es de uso fácil e intuitivo en la creación de interfaces con transiciones.
- Se pueden realizar mejoras en las aplicaciones sin tener que modificar el código fuente ni recompilarlo, ya que este está incrustado y probado en la biblioteca.
- Las aplicaciones diseñadas ya son capaces de aportar enriquecimiento visual aceptable para los usuarios.

5. REFERENCIAS BIBLIOGRÁFICAS

- iPod ABC. (2007). "Configurar un pase de diapositivas". <http://docs.info.apple.com/article.html?artnum=304679-es>.
- Pérez, M. (2007). "Animación y efectos". <http://roble.pntic.mec.es/~mperez8/pagweb3/animacion.htm>.
- Bravo R.J. (2002). "Elaboración de presentaciones con Ordenador". Universidad Politécnica de Madrid. <http://www.ice.upm.es/av/html/TecnoRec/Temario/PRESENdib02.pdf>.
- "Efectos gráficos", (2007). <http://grafics.pina.cat/definicio.php?id=21>.
- "Ingeniería de Sistemas y Computación", (2004). Grupo de Desarrollo en Allegro, GDA. Universidad Tecnológica de Pereira. http://gda.utp.edu.co:9673/gda/documentacion/programacion_3d/directx/tutorial08.Libros

Authorization and Disclaimer

Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.