

# Diseño asistido por ordenador

4ª Curso Ingeniería Informática

J.C. Torres  
Dpt. Lenguajes y Sistemas Informáticos  
ETS. Ingeniería Informática  
Universidad de Granada

## **TEMA 3. Interacción**

Este tema aborda las técnicas de interacción útiles para crear o editar el modelo geométrico.

## Índice

- 3.1. Introducción
- 3.2. Fundamentos
- 3.3. Dispositivos lógicos
- 3.4. Modos de entrada
- 3.5. Técnicas de Interacción
- 3.6. Entrada de transformaciones geométricas
- 3.7. Software de base para sistemas interactivos

## 1. Interacción

Al editar el modelo geométrico es necesario introducir coordenadas de puntos, especificar transformaciones geométricas y seleccionar componentes del modelo. Estas funciones podemos realizar con un texto, usando ordenes en algún lenguaje que el sistema entienda e interprete para generar el modelo geométrico. Pero este método es poco efectivo, y limita la capacidad de actuación del usuario. Por ejemplo, se puede realizar todas las operaciones de entrada usando un teclado, y sin ninguna técnica especial, introduciendo las coordenadas numéricamente y referenciando los componentes de un modelo geométrico asignándole un identificador a cada uno, pero esto es ineficiente y complejo, pues obliga al usuario a aprenderse los identificadores.

Las técnicas de interacción gráfica son el soporte de la entrada de información geométrica del sistema de Diseño. Entre estas, las técnicas de posicionamiento y selección poseen una especial relevancia. Las técnicas de posicionamiento se utilizan para la introducción de posiciones 2D o 3D, esenciales para construir elementos geométricos. Las técnicas de selección permiten la identificación interactiva de un componente del modelo, son por tanto esenciales para la edición.

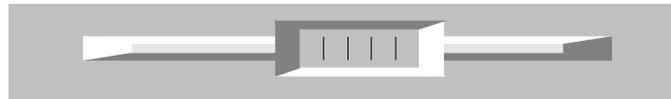


Figura 3.1 Representación de un dial

### Complementos 3.1. Interacción en informática gráfica

Actualmente, tan solo los sistemas gráficos que realizan procesos muy costosos en tiempo de cálculo, son no interactivos, por ejemplo generación de imágenes realista mediante cálculo de radiosidades, o generación de animaciones. El desarrollo de los primeros sistemas gráficos interactivos se remonta a los orígenes de la Informática Gráfica [Foley 90], sin embargo, su utilización no se generalizó hasta el desarrollo comercial de dispositivos de entrada flexibles y de los primeros estándares gráficos [Myers 77].

Podemos decir que un *sistema gráfico es interactivo* si su operación se realiza siguiendo ciclos cortos de procesamiento, en cada uno de los cuales el usuario puede indicar al sistema la siguiente tarea a realizar. El usuario decide la operación a realizar en base a la información mostrada por el sistema.

El usuario introduce la información por medio de dispositivos de entrada. Estos pueden ser de propósito general (teclado), o específicos para la generación de datos geométricos (digitalizador). Podemos clasificar los dispositivos de entrada atendiendo a la información que procesan de forma directa, en: alfanuméricos, localizadores, valuadores y de elección (tabla 3.1).

No obstante, siempre es posible generar cualquier tipo de información con cualquier dispositivo de entrada, simulando, con ayuda de información de realimentación mostrada en pantalla, el funcionamiento de otro dispositivo. Así, por ejemplo, es habitual realizar la entrada de valores numéricos usando dispositivos localizadores que actúan sobre la representación gráfica de un dial (ver figura 3.1).

Tipo de Dispositivo	Tipo de información	Dispositivos
Alfanuméricos	Texto	Teclado
	Posiciones 2D	Tableta digitalizadora Lápiz óptico Pantalla táctil
Localizadores	Posiciones 3D	Digitalizador 3D
		Guantes
	Desplazamientos 2D	Ratón TrackBall Joystick
De valores	Magnitudes reales	Dial
De opciones	Enumerado	Panel de teclas Teclas de función

Tabla 3.1 Dispositivos de entrada

## 2. Fundamentos

Existen distintas técnicas útiles para la implementación de funciones de entrada. La mayor parte de estas están basadas en la utilización de información de *realimentación*. La *realimentación* es el mecanismo mediante el cual el sistema da información al usuario útil para determinar la siguiente acción a realizar. La información de realimentación que el sistema genera en cada momento depende lógicamente del estado del sistema y de la información previamente entrada por el usuario.

En general, la información de realimentación muestra al usuario los pasos seguidos por el sistema, lo que el sistema puede hacer en cada momento, y como va a interpretar las acciones que el usuario pueda realizar. La realimentación es un ingrediente esencial en todas las funciones de entrada de cualquier sistema interactivo. Esta se puede usar con diferentes fines específicos: mostrar el estado del sistema; como parte de una función de entrada; o simplemente para reducir la incertidumbre del usuario [Salesin 93].

## 3. Dispositivos lógicos.

La dependencia de la aplicación con el hardware de entrada se resuelve, en parte, trabajando con dispositivos abstractos, denominados *dispositivos lógicos*. Cada dispositivo lógico representa un dispositivo físico ideal, que genera un único tipo de información de entrada. Cada dispositivo físico puede ser tratado como una realización de uno, o varios, dispositivos lógicos [Foley 74]. Esta construcción se ha usado en prácticamente todas las librerías gráficas, ya que garantiza el que las funciones de entrada sean independientes de los dispositivos. La aplicación debe indicar que necesita leer una posición, y el usuario actual sobre uno de los dispositivos que pueden generar posiciones. El dispositivo concreto usado es irrelevante tanto para la aplicación como para el usuario. Entre los dispositivos lógicos habituales en sistemas gráficos, se encuentran dispositivos para leer valores numéricos, texto, opciones, identificadores de objetos definidos por el usuario y posiciones.

La lectura de posiciones y la selección son características de los sistemas gráficos. El dispositivo de lectura de posiciones, conocido como localizador (*locator*), representa un digitalizador, sobre el que el usuario puede indicar posiciones, que son entregadas a la aplicación como coordenadas 2D. El dispositivo posee su propio

sistema de coordenadas, por lo que es necesario realizar una transformación al sistema de coordenadas usado en la aplicación.

El dispositivo de selección, *Pick*, lee el identificador de un componente del modelo geométrico. Es esencial para permitir la edición en cualquier sistema gráfico. Si bien no existe ningún dispositivo físico que directamente genere esta información, podemos considerar que el comportamiento se asemeja al de un lápiz óptico. La implementación de esta operación requiere la consulta del modelo geométrico [ver capítulo 5]. Habitualmente se implementa realizando una lectura de posición, y relacionando el punto obtenido con el modelo geométrico ('correlate'). Esta relación se puede realizar de varios modos, uno de los más simples es realizar una visualización del modelo, para determinar que componente genera el pixel de la imagen sobre el que esta el cursor (ver tema 2, sección 2)

### **Complementos 3.2. Otros dispositivos lógicos en PHIGS**

Un *valuator* lee valores numéricos. El dispositivo típico de entrada es el dial, pero se puede asociar tanto a dispositivos alfanuméricos como localizadores, haciendo uso de técnicas adecuadas y de información de realimentación (figura 3.1).

El dispositivo *text* lee una cadena de caracteres, el prototipo de dispositivo es el teclado. Su realización con otro tipo de dispositivos da lugar a formas de interacción complejas.

El dispositivo pulsador, u opción (*choice*), lee el identificador de una alternativa en un menú, que se elige entre varias posibles. Típicamente la opción representa una orden que el usuario da al sistema. El dispositivo característico es el panel de teclas, o las teclas de función de un teclado. No obstante, se suele realizar mediante dispositivos localizadores, mostrando en pantalla botones que representan cada una de las opciones. Estos dispositivos, o las operaciones de entrada que realizan, aparecen en cualquier sistema interactivo.

Para realizar un locator mediante dispositivos que generen desplazamientos (p.e. un ratón), se hace uso de realimentación, mostrando en pantalla la posición en que se encuentra el dispositivo lógico. La imagen de la posición del dispositivo en pantalla, que habitualmente es una cruz o una flecha, se denomina *cursor*. El sistema interpreta los desplazamientos del dispositivo como desplazamientos de la posición del cursor. La posición leída será la posición en que se encuentra el cursor. En este caso, el sistema de coordenadas del dispositivo lógico será el de la pantalla. Por tanto, la transformación que se debe realizar es la inversa de la transformación de visualización. Haciendo uso del mismo tipo de realimentación, se puede representar un localizador mediante cualquier dispositivo de entrada que genere al menos cuatro valores distintos.

## **4. Modos de entrada.**

Otro de los problemas relacionados con las operaciones de entrada es el de sincronizar las acciones del usuario con la aplicación. Hay situaciones en las que el usuario puede anticiparse a la operación de lectura, y otras en las que esto no se debe permitir. Por ejemplo, cuando la aplicación debe mostrar previamente información necesaria para la operación, o cuando el tipo de información que se puede introducir depende del proceso que está llevando a cabo la aplicación.

Por otra parte, hay situaciones en las que el usuario debe generar datos individuales, cuya generación se indica con alguna marca de validación (por ejemplo la pulsación de un botón del ratón). En otros casos se deben introducir datos de modo continuo, muestreando el dispositivo; por ejemplo cuando se digitaliza una curva [Hopgood 92].

Estos problemas de sincronización se resuelven, en los sistemas gráficos, haciendo también uso de mecanismos de abstracción. A las abstracciones del comportamiento se les denomina *modos de entrada*. En cada momento, la aplicación puede asociar un modo a cada dispositivo lógico. Habitualmente se definen tres *modos* de funcionamiento: petición, suceso y muestreo.

### Complementos 3.3. Modos de entrada en PHIGS

El *modo petición* ('request') funciona como una lectura sin buffer de entrada. El usuario no puede generar datos hasta que la aplicación los solicita, y ésta espera hasta que el usuario finaliza la operación. En el *modo suceso* ('event') se introduce un buffer, de modo que el usuario puede realizar entradas antes de que la aplicación las solicite. En este modo la aplicación puede no quedar suspendida, si el usuario ha introducido algún dato previamente. En estos dos modos se puede hablar de operaciones de entrada individuales, que el usuario finaliza con alguna acción específica sobre el dispositivo, y cada una de las cuales genera un dato de entrada.

El *modo muestreo* ('sample') caracteriza un funcionamiento distinto, que no está basado en la realización de operaciones individuales, sino en la asociación de un valor al dispositivo. El usuario puede modificar el valor actuando sobre el dispositivo, y la aplicación puede consultar su valor. El funcionamiento de este modo es asíncrono.

En el modo petición el software de entrada puede activarse desde la aplicación. Es decir, puede verse como un procedimiento. En el modo muestreo, debe activarse cuando se producen cambios en los dispositivos, modificando su medida, que se almacena internamente, y que se podrá consultar desde la aplicación. En el modo suceso, el software de entrada es activado también por el dispositivo. En este caso se almacena una cola de sucesos, de la que el programa puede extraer elementos. De cada suceso se debe almacenar el dispositivo lógico que lo produjo y su medida, o valor.

## 5. Técnicas de interacción

Con frecuencia, el usuario debe introducir una posición que satisfaga una determinada condición, por ejemplo tener la misma coordenada x que la posición previa. En estas situaciones, la aplicación puede ayudar al usuario a realizar su tarea proporcionándole técnicas específicas que simplifiquen el proceso de interacción. En el ejemplo anterior se puede hacer que el cursor solo se mueva en horizontal.

En general, el diseño de software de entrada debe tener en cuenta el contexto concreto en el que va a utilizar, tratando de dar la mejor asistencia posible al usuario para la función que este debe realizar. En esta sección se examinan algunas de las técnicas de posicionamiento y selección más usadas.

### 5.1. Técnicas de posicionamiento

El objetivo de las operaciones de posicionamiento es permitir la introducción de un punto. Desde el punto de vista del usuario, que utiliza un dispositivo que controla un cursor, el proceso de entrada es el que se muestra en la figura 3.2.a.

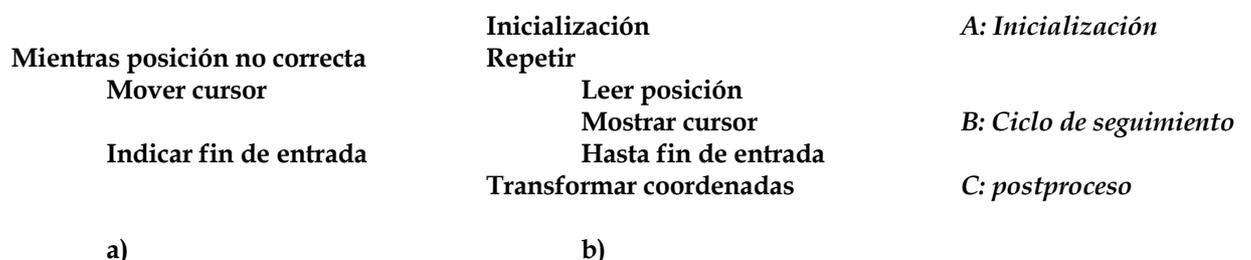


Figura 3.2 Posicionamiento. Desde el punto de vista del usuario (a), y del software de entrada (b).

El proceso en el software de tratamiento es el que se muestra en la figura 3.2.b. Se observa una estructura cíclica, que se corresponde con el ciclo de búsqueda que realiza el usuario. En este proceso podemos distinguir tres etapas: la inicialización (A), el ciclo de seguimiento (B), y el postproceso (C).

En la *inicialización* se muestra el eco del dispositivo, y se inicializa el hardware de entrada. En el *ciclo de seguimiento* se leen entradas del dispositivo físico, actualizando el eco. En el *postproceso* se realiza las transformaciones necesarias para convertir el dato generado por el dispositivo al formato requerido por la aplicación. En este punto se realiza la conversión de sistema de coordenadas.

Se pueden incorporar técnicas en cada uno de estas etapas. Antes del ciclo de seguimiento se puede representar en pantalla **información previa**, que sirva de *guía* al usuario. Una de las técnicas más conocidas de este tipo, es la utilización de rejillas. Una *rejilla* es una cuadrícula, o una distribución regular de puntos, que se dibuja en la pantalla, para indicar de forma aproximada la situación del cursor. Las *escalas* o reglas son también información previa. La presentación de información previa puede hacerse para más de una operación de entrada consecutiva, y no requiere demasiados recursos.

En el **postproceso**, después del ciclo de seguimiento, se pueden hacer modificaciones en la posición obtenida, para que se satisfaga alguna *restricción*. Tal vez la restricción más utilizada sea la *modular*, que trunca las coordenadas introducidas según un determinado espaciado. Ésta se suele utilizar en combinación con una rejilla. Otra restricción simple es la *direccional*, usada para la introducción de líneas. La restricción direccional modifica el punto introducido para que la línea que lo une al anterior tenga una pendiente fija, normalmente horizontal o vertical. La restricción *gravitacional* proyecta el punto introducido en el modelo geométrico. Se puede diseñar de forma que devuelva un punto que sea vértice del modelo geométrico, o que esté sobre un componente de este. En cualquier caso, su implementación implica realizar una búsqueda en el modelo geométrico, es decir una operación de selección. El usuario debe ser consciente de la utilización de restricciones en la aplicación, y debe ser informado de la modificación realizada en el punto introducido.

En el ciclo de seguimiento se suelen utilizar técnicas basadas tanto en la presentación de información de realimentación como de transformación de la información entrada por el usuario. La técnica más conocida de las usadas en el ciclo de **seguimiento** es la *línea elástica*, utilizada para dibujar líneas, u otro tipo de elementos geométricos. Ésta técnica consiste simplemente en dibujar una línea entre el cursor y un punto de referencia, normalmente el último introducido. La imagen de la línea cambia al modificarse la posición del cursor, y por tanto debe hacerse dentro del ciclo de seguimiento. Cada cambio de posición implica borrar la línea actual, y dibujar la línea a la nueva posición. Esto debe hacerse manteniendo el resto del dibujo. Si se dibuja la línea en modo Xor, el borrado se realiza como un redibujado en Xor, que restaura la imagen original.

La modificación de la línea debe hacerse cada vez que el cursor cambia de posición, y en todo caso, no menos de 5 veces por segundo mientras el cursor se mueva. También pueden implementarse las restricciones geométricas en el ciclo de seguimiento, con lo que el usuario puede saber la posición que se tomará antes de finalizar la operación. La única restricción, para implementar las restricciones en el ciclo de seguimiento, es la velocidad de procesamiento del ordenador, que puede hacer imposible la implementación de técnicas costosas, como las restricciones gravitacionales.

Tanto la presentación de información previa, como las restricciones, se pueden realizar en la aplicación, aun en el caso de que se utilice una librería de entrada, pues no afectan al ciclo de seguimiento. La incorporación de técnicas en el ciclo de seguimiento solo se puede realizar en el software de entrada.

## 5.2. Técnicas de selección

La selección permite al usuario referenciar componentes de un modelo geométrico. Es por tanto esencial en cualquier aplicación gráfica que requiera la edición de un modelo. El proceso de selección suele realizarse como un posicionamiento y una búsqueda en el modelo geométrico. Se pueden definir diversas estrategias de búsqueda, la más simple conceptualmente es buscar el elemento más próximo al punto introducido. Este criterio es fácil de implementar para modelos geométricos 2D sencillos, pero es costoso para modelos complicados. Una forma de simplificarlo es la utilización de puntos de selección, que funcionan como únicos puntos a comprobar al hacer la selección.

Una alternativa es hacer uso de algoritmos contenidos en otros componente del sistema gráfico, como el módulo de visualización o de recorte. Si se utiliza como criterio de selección que el objeto **genere la imagen en el punto donde se encuentra el cursor**, se puede determinar el objeto seleccionado reproduciendo el proceso de visualización. Alternativamente, se puede utilizar como criterio que el objeto esté **contenido en una ventana**, que o bien se construye automáticamente a partir de la posición del cursor, o bien genera explícitamente el usuario. En este caso, el algoritmo para localizar el objeto seleccionado es el de recortado. Estas técnicas se pueden usar tanto para selección 2D como 3D.

Independientemente de cual sea el criterio de selección, podrán aparecer ambigüedades, tanto por la naturaleza del criterio como por la utilización que el usuario hace de él. Decimos que hay una *ambigüedad* en la selección cuando el criterio utilizado encuentra más de un objeto. Esto puede ocurrir con el criterio de distancia, y más frecuentemente con el de inclusión. También puede aparecer ambigüedad por una mala 'evaluación' del criterio por parte del usuario, que entiende que el objeto que se debería haber seleccionado es otro. La solución habitual es hacer uso de realimentación, para pedir confirmación al usuario, o pedirle que elija entre los posibles candidatos. Concretamente, se puede mostrar una secuencia de posibles elecciones, de entre las cuales el usuario elige una. El objeto seleccionado se resalta, cambiando su brillo, color, estilo, parpadeando o enmarcándolo, y se le pide al usuario que confirme la selección realizada.

Cuando se utilizan modelos geométricos jerárquicos, es necesario saber a que nivel concreto de la jerarquía se desea realizar la selección. Piénsese, por ejemplo, en un modelo formado por líneas, estructurado en dos niveles. En el nivel superior almacenamos las líneas con referencias a sus vértices, que se encuentran en el nivel inferior. Si el usuario hace una selección cerca de un extremo, no sabemos si desea seleccionar la línea o el punto. Este problema, conocido como problema de *amplitud*, se puede resolver especificando el nivel al que se desea seleccionar. Esto puede hacerse utilizando ordenes distintas para cada nivel, o especificando el nivel previamente, o posteriormente, a la selección. En una estructura multinivel homogénea se puede implementar un mecanismo para moverse por la estructura, haciéndose la selección al nivel actual.

### 5.3. Posicionamiento 3D

La forma natural de introducir posiciones 3D es mediante dispositivos físicos 3D. No obstante, aunque el uso de guantes ('dataglove') se está extendiendo, gracias a la difusión de la realidad virtual, actualmente su uso en aplicaciones gráficas generales está bastante limitado.

La introducción de posiciones 3D, utilizando dispositivos de entrada 2D, requiere el uso de técnicas especiales. Una de las estrategias posibles es hacer que el usuario introduzca las tres coordenadas del punto actuando sobre dos o tres proyecciones. La figura 3.3.a muestra la utilización de tres vistas, observándose que las coordenadas de cada vista están ligadas a las de las vistas contiguas. Este mecanismo solo permite generar puntos en un octante. Por otra parte, la introducción de un punto se debe descomponer en varios posicionamientos 2D.

Estas limitaciones se pueden eliminar si se utiliza un cursor 3D, que se visualiza sobre una vista tridimensional del modelo. Esto plantea dos problemas fundamentales, establecer una correspondencia entre los movimientos del dispositivo 2D y el cursor 3D y la representación del cursor en sí. La representación del cursor está condicionada por la capacidad del ordenador, ya que se debe de regenerar cada vez que se actúa sobre el dispositivo. Entre las formas más simples está el dibujo de las proyecciones del cursor en los planos del sistema, o la representación del cursor como tres rectas ortogonales. (ver figura 3.3.b). Con equipos que realizan la visualización en tiempo real se pueden usar cursores más sofisticados, como el propuesto por Scott Hudson, consistente en un objeto sólido, que se visualiza con el resto del modelo, proyectando sombras sobre él [Hudson 92].

Existen varias técnicas para generar los desplazamientos en las tres direcciones, a partir de los del dispositivo 2D. La más simple es utilizar un botón, o tecla, para conmutar la interpretación del movimiento del dispositivo, del plano X-Y al X-Z.

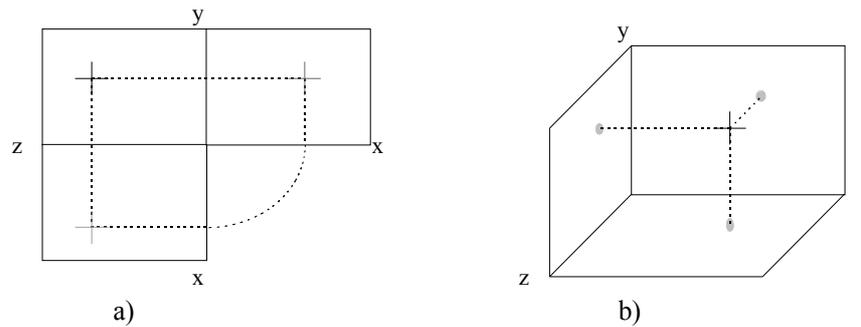


Figura 3.3. Introducción de puntos 3D.

## 6. Entrada de transformaciones geométricas

La edición de modelos geométricos requiere la introducción tanto de puntos como de transformaciones geométricas. Las transformaciones geométricas utilizadas en modelado se pueden definir a partir de las coordenadas de puntos de referencia. Concretamente, una traslación, o un escalado, se pueden definir a partir del vector que une dos puntos. Una rotación 2D se puede especificar a partir de la componente angular de un punto. Una rotación 3D, respecto a un eje de coordenadas, se puede especificar de la misma forma, usando como referencia la componente angular del punto en el plano perpendicular al eje [Emmerik 90, Glassner 90].

Utilizando la técnica de líneas elásticas, la introducción de los puntos de referencia se puede hacer de forma transparente para el usuario. En este caso, la transformación geométrica se calcula y aplica al componente del modelo a transformar dentro del ciclo de seguimiento. De esta forma el usuario tiene la impresión de que el cursor *arrastra*, o deforma, al objeto. Este tipo de técnicas comenzaron a usarse tempranamente en sistemas vectoriales, en los que la naturaleza del proceso de refresco facilitaba la edición interactiva del modelo.

Para la introducción de rotaciones 3D se han definido técnicas específicas. Entre las más conocidos está la de Evans, Tanner y Wein [Evans 81]. Esta técnica simula la rotación de una bola, distinguiendo los movimientos lineales del cursor, de los circulares. Los primeros se interpretan como rotaciones según los ejes X o Y. Los movimientos circulares del cursor se corresponden con rotaciones según el eje Z, perpendicular al plano de movimiento del cursor.

## 7. Software de base para sistemas interactivos

Actualmente existen gran cantidad de librerías y herramientas que ayudan a la implementación de sistemas gráficos interactivos de distintas formas. Los servicios básicas los ofrecen los sistemas de gestión de ventanas, que se encargan de gestionar la pantalla y los dispositivos de entrada. Sobre estos funcionan las librerías gráficas, que aportan funciones gráficas de salida, en algunos casos también de creación y edición de modelos geométricos, y dan una visión más abstracta de los dispositivos de entrada y salida.

Sobre estos elementos, se suelen montar librerías de técnicas de interacción, que permiten realizar operaciones de entrada complejas, encapsulando funciones de entrada y realimentación. Por ejemplo, lectura de parámetros de un cuadro de diálogo. La librería, o *toolkit*, usada determina la apariencia de la interfaz de la aplicación. Los toolkit suelen permitir la conexión de elementos de la interfaz con procedimientos de la aplicación, haciendo que el procedimiento sea invocado cuando se selecciona el elemento de la interfaz. Este mecanismo de llamada de la interfaz a la aplicación se denomina *callback*.

La figura 3.4 muestra las relación entre los distintos elementos software. El software de aplicación puede hacer uso de servicios de cada uno de los niveles. Esto, no obstante, puede generar modificaciones en el comportamiento de funciones de otros niveles.



Figura 3.4. Estructuración del software

## Bibliografía

- Emmerik90 Emmerik M.J.G.M. van: "A Direct Manipulation Technique for Specifying 3D Object Transformations with a 2D Input Device". **Computer Graphics Forum**. Vol. 9, N.4, Dec. 1990. pp. 355-362.
- Foley82 J. Foley, V. Wallace: "The Art of Natural Graphic Man-Machine Conversation". *Proceeding of the IEEE*, 1974. Reproducido en Beatty, Botch (Ed). *Tutorial: Computer Graphics*, IEEE Press, 1982.
- Foley90 Foley J.D.; van Dam A.; Feiner S.K.; Hughes J.F.: *Computer Graphics. Theory and Practice*. Addison-Wesley 1990.
- Glassner90 Glassner A.S.: "A Two dimensional View Controller". **ACM Transactions on Graphics**. Vol.9, N.1, Jan. 1990. pp. 138-141.
- Hear94 Hearn D.D.; Baker M.P.: **Gráficas por computadora**. Prentice Hall 1994
- Hopgood92 Hopgood F.; Duce D.A.; Johnston D.J.: *A Primer for PHIGS. C programmers' Edition*. John Wiley & Sons. 1992.
- Hudson92 S.E. Hudson: "Adding shadows to a 3D Cursor". **ACM Transactions on Graphics**. Vol. 11, N.2, pp. 193-199. 1992.
- Myers82 W. Myers: "Interactive Computer Graphics: Poised or Takeoff". **ACM Computer Graphics**. Proceeding of the 4th Conference on Computer Graphics & Interaction Techniques 1977. Reproducido en Beatty, Botch (Ed). *Tutorial: Computer Graphics*, IEEE Press, 1982
- Newman81 Newman W.M.; Sproull R.F.: *Principles of Interactive Computer Graphics*. McGraw-Hill 1981
- Nielson87 G. Nielson, D. Olsen: "Direct Manipulation Techniques for 3D Objects using 2D Locator Devices". *ACM workshop on Interactive 3D Graphics*. pp.175-182. 1987
- Salesin93 Salesin C.; Barzel R.: "Adjustable Tools: An Object-Oriented Interaction Metaphor". **ACM Transactions on Graphics**. Vol.12, N.1, Jan. 1993. pp. 103-107.
- Salmon87 Salmon, R.; Slater, M.: *Computer Graphics: Systems and Concepts*. Addison Wesley, 1987.
- Sproull86 Sproull R.; Southerland: *Device independent Graphics*. McGraw-Hill 1986

## Ejercicios

1. Dar un ejemplo de situaciones en las que sea preferible utilizar cada uno de los dispositivos lógicos de entrada.
2. Dar un ejemplo de situaciones en las que sea preferible utilizar cada uno de los modos de entrada.
3. Escribe un procedimiento para la introducción de líneas utilizando la técnica de la línea elástica. ¿Puede aumentarse la velocidad sin renunciar a la interacción?
4. Describir el proceso de conversión de coordenadas de dispositivo a coordenadas del mundo.
5. Realizar una lista de dispositivos de entrada que se pueden utilizar en cada tipo y modo de entrada.
6. ¿Es posible implementar el dispositivo lógico STRING utilizando como dispositivo físico un ratón? ¿Porqué / cómo?
7. Realizar en pseudocódigo una restricción gravitacional.
8. Supongamos un sistema gráfico que dispone tan sólo de entrada en modo *sample*. ¿Es posible crear una librería de funciones que, utilizando las funciones de entrada de dicho sistema gráfico, implemente entrada en modo *event*? ¿Porqué / como?. Realizar una implementación en pseudocódigo para el ratón.
9. Redactar un algoritmo que permita editar un rectángulo (que podría ser el marco de un viewport) usando el *locator*. Se utilizarán dos puntos para seleccionar la operación a realizar. El punto central del borde superior permite cambiar la posición y la esquina inferior izquierda el tamaño. Al pulsar el botón izquierdo estando sobre alguno de los puntos de selección, se entra en el proceso de edición, del que se sale al soltarlo.
10. Describir cómo se podría realizar la interacción con un ratón, de tal forma que el cursor se moviese sobre un rectángulo, haciendo que cuando salga por un lado, aparezca por el lado opuesto.
11. Implementar en OpenGL un método para introducir una poligonal.
12. Implementar en C el procedimiento que permitiría realizar un resaltado, cambiando el color de las aristas, en la siguiente estructura. Mostrar la evolución de la estructura y la del puntero.

```

pset_local_trans3(m1,PTYPE_PRECONCAT);
pexec_struct();
pset_local_trans3(m2,PTYPE_POSTCONCAT);
pexec_struct();

```

13. Realizar un algoritmo (cíclico) que permita solapar ventanas (haciendo un click de ratón sobre la misma) y moverla (si nos situamos en el punto superior izquierdo de la misma, arrastrando hasta dejar de pulsar el ratón).  
NOTA: Si se mueve una ventana y no es visible, pasará a ser la activa.
14. Desarrollar un algoritmo que permita manipular mediante el *locator* la forma de un rectángulo. Al pulsar el botón izquierdo y estando sobre alguno de los puntos de selección, moverlo, y cuando se suelte, se actualizará el tamaño del mismo (ver figura 2.22). La esquina inferior modifica el tamaño, y la superior la posición.