

I.E.



**Industria y Comercio**

SUPERINTENDENCIA

SOLICITUD  
PATENTE DE INVENCION

EXPEDIENTE N°

**05-21195**

TITULO: MODELO PARA DEFINIR MARCOS  
CLAVE (KYFRAMING) CON ATRIBUTO DE OBJETO  
COMBINADO

CLASIFICACION INTERNACIONAL: 6 06 F 17/60

SOLICITANTE: MICROSOFT CORPORATION

DOMICILIO: REDMOND, WASHINGTON, ESTADOS  
UNIDOS DE AMERICA

APODERADO: DARIO CARDENAS NAVAS

BOGOTÁ, D.C. \_\_\_\_\_

# PETITORIO

**Industria y Comercio**  
**SUPERINTENDENCIA**

SUPLENIMIENTOS Y CORRECCIONES  
Radicación : 05021195 00000000 Folios: 31  
Fecha (MM/AA): 2005-03-08 12:27:22  
Trámite : 002 PRIORITA Y 1 REGISTRO/ 411 PRESENTE  
Dependencia: CECU DIVISION DE MARCAS Y DISEÑOS

## FORMULARIO UNICO DE SOLICITUD DE PATENTE 2000-3

1

### SOLICITUD DE:

☒

Patente de Invención

☐

Patente de Modelo de Utilidad

**SOLICITANTE**  
(71)

Nombre: **MICROSOFT CORPORATION**  
Dirección: **One Microsoft Way Redmond, Washington 98052, Estados Unidos de America**  
Teléfono: **3123600** Fax: **3122410**  
E-Mail: **general@cardenasycardenas.com**  
Lugar de Constitución: **Estado de Washington, Estados Unidos de América**  
Domicilio: **REDMOND, WASHINGTON, ESTADOS UNIDOS DE AMERICA**

### IDENTIFICACION

C.C. ☐ NIT ☐  
C.E. ☐ Otro ☒  
Número \_\_\_\_\_

**REPRESENTANTE  
O APODERADO**  
(74)

Nombre: **DARIO CARDENAS NAVAS**  
Dirección: **Carrera 7 No 71-52 Torre B Piso 9**  
Teléfono: **3123600** Fax: **3122410**  
E-Mail: **general@cardenasycardenas.com**

### IDENTIFICACION

C.C. ☒ NIT ☐  
C.E. ☐ Otro ☐  
Número **17.068.629 BTA**  
TP **1.250 C.S.J.**

**INVENTOR(ES)**  
(72)

Nombre:  
1. Devon L Strawn  
2. Kenneth L. Young  
3. Kart B Jacob  
4. Patrick G Mullen  
5. Peter Faraday  
Dirección:  
1. One Microsoft Way Redmond, Washington 98052, Estados Unidos de América  
2. One Microsoft Way Redmond, Washington 98052, Estados Unidos de América  
3. One Microsoft Way Redmond, Washington 98052, Estados Unidos de América  
4. One Microsoft Way Redmond, Washington 98052, Estados Unidos de América  
5. Reino Unido  
Teléfono: **3123600** Fax: **3123600**  
E-Mail: **general@cardenasycardenas.com**  
Domicilio:  
1. REDMOND, WASHINGTON, ESTADOS UNIDOS DE AMERICA  
2. REDMOND, WASHINGTON, ESTADOS UNIDOS DE AMERICA  
3. REDMOND, WASHINGTON, ESTADOS UNIDOS DE AMERICA  
4. REDMOND, WASHINGTON, ESTADOS UNIDOS DE AMERICA  
5. REINO UNIDO

### IDENTIFICACION

C.C. ☐ NIT ☐  
C.E. ☐ Otro ☒

Título(54):

5

**MODELO PARA DEFINIR MARCOS CLAVE (KEYFRAMING) CON ATRIBUTO DE OBJETO COMBINADO**

6

Prioridad

SI ☒

NO ☐

(33) País de Origen

U.S.A.

(32) Fecha

Abril 15 de 2004

(31) N° de Solicitud

10/825035

Para publicar a partir de la fecha de la presente solicitud a los:

☐ 6 meses ☐ 12 meses ☐ 18 meses ☐ Otro

*Clasificación Internacional*

*G06F 19/00*

Comprobante de Pago No.:

05-14,522 / 05-14,528

Feb 24, 2005

2

# ANEXOS

- ☒ Comprobante de pago de la tasa de presentación de la solicitud. (No. )
- ☐ Comprobante de pago de la tasa por concepto de excedente de palabras en la publicación.
- ☒ Comprobante de pago de la tasa por concepto de reivindicación de prioridad. (No.)
- ☒ Poderes, si fuere el caso. (Protocolo No 16.069)
- ☒ Certificado de existencia y representación legal cuando el solicitante sea persona jurídica. Protocolo No 16.069)
- ☐ Copla certificada de la primera solicitud, si se reivindica prioridad.
- ☐ Traducción simple de la primera solicitud, si se reivindica prioridad.
- ☐ Documento de cesión del inventor al solicitante o a su causante.
- ☒ Descripción de la invención.
- ☒ Una o más reivindicaciones.
- ☒ Dibujos y/o planos necesarios.
- ☐ De ser el caso, copia del contrato de acceso.
- ☐ De ser el caso, documento que acredite la licencia o autorización de uso de conocimientos tradicionales de la comunidad de las comunidades indígenas.
- ☐ De ser el caso, certificado de depósito del material biológico.
- ☒ Arte final 12x12 y 6x6 cm.
- ☒ Resumen y extracto para publicación.

10

NOMBRE:

DARIO GARDENAS NAVAS



FIRMA:



C.C. 17.086.829 BTA

TP. 1.250 C.SJ.

SUPERINDUSTRIA Y COMERCIO  
Modificacion : 50021150 00000000 Folio: 61  
Fecha (VER): 2005-02-04 12:12:22  
(Tramite : 002 PATENTES D 1 REGISTRADO/ 1 PRESENTAC  
Dependencia: 2020 DIVISION DE NUEVAS CREACIONES

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

NIT : 800176089-2

RECIBO OFICIAL DE CAJA : 05 - 14,522  
FECHA : FEBRERO 24 DE 2005

\*\*\*\*\* CONSIGNACION \*\*\*\*\*

| DEPOSITANTE                           | TIPO PAGO | BANCO         | CUENTA      | No. PAGO | FECHA PGO  | Vr. PAGO     |
|---------------------------------------|-----------|---------------|-------------|----------|------------|--------------|
| CARDENAS Y CARDENAS P.I. CONSIGNACION |           | BANCO POPULAR | 050-00110-6 | 37960513 | 24/02/2005 | 8,412,000.00 |

\*\*\*\*\* CONCEPTO \*\*\*\*\*

| CANT. RENTISTICO | CONCEPTO | TOTAL CONCEPTO |
|------------------|----------|----------------|
|------------------|----------|----------------|

1 50005-01-01 SOLICITUDES

1 TRAMITES DE SOL. DE PATENTE DE IN  
TOTAL :

443,000.00  
443,000.00

SON: CUATROCIENTOS CUARENTA Y TRES MIL PESOS

RESPONSABLE : \_\_\_\_\_

CARDENAS & CARDENAS  
ABOGADOS

RECIBO DE CAJA APLICADO AL EXPEDIENTE No. MODELO PARA DEFINIR MARCOS CLAVE  
(KEYFRAMING) CON ATRIBUTO DE OBJETO  
COMBINADO

A

SUPERINDUSTRIA Y COMERCIO  
Radicación : 05021195 00000000  
Fecha (AMB): 2005-02-04 12:12/122  
Transito : 002 PATENTES # 1 NEGOTIUM/ 411 INGENIERIA  
Dependencia: CNEC DIVISION DE NUEVAS CREACIONES  
Folios: 01

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

NIT : 800176089-2

RECIBO OFICIAL DE CAJA : 05 - 14,528  
FECHA : FEBRERO 24 DE 2005

\*\*\*\*\* CONSIGNACION \*\*\*\*\*

| DEPOSITANTE                           | TIPO PAGO | BANCO         | CUENTA      | No. PAGO | FECHA PGO  | Vr. PAGO     |
|---------------------------------------|-----------|---------------|-------------|----------|------------|--------------|
| CARDENAS Y CARDENAS P.I. CONSIGNACION |           | BANCO POPULAR | 050-00110-6 | 37960513 | 24/02/2005 | 8,412,000.00 |

\*\*\*\*\* CONCEPTO \*\*\*\*\*

| CANT. RENTISTICO          | CONCEPTO                             | TOTAL CONCEPTO |
|---------------------------|--------------------------------------|----------------|
| 1 50005-01-01 SOLICITUDES | 82 INVOCACION DE UNA PRIORIDAD EN NU | 124,000.00     |
|                           | TOTAL :                              | 124,000.00     |

SOM: CIENTO VEINTICUATRO MIL PESOS

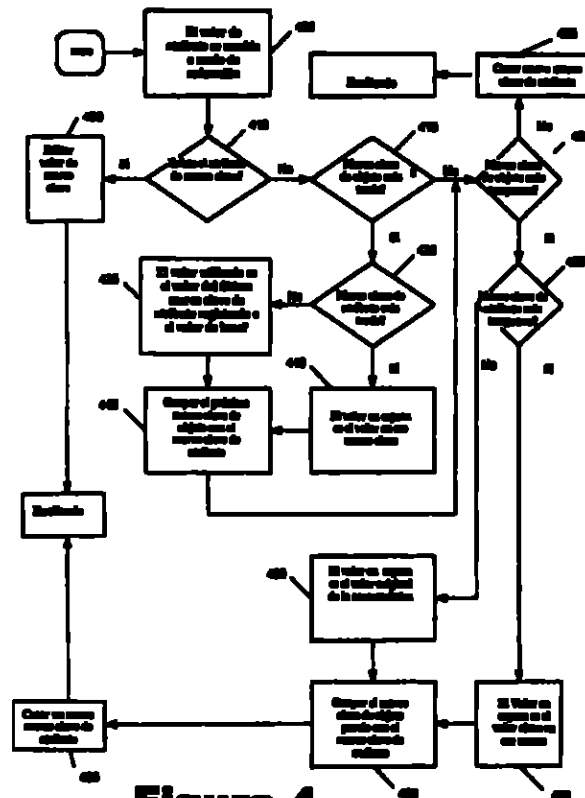
RESPONSABLE : \_\_\_\_\_

CARDENAS & CARDENAS

ABOGADOS

RECIBO DE CAJA APLICADO AL EXPEDIENTE No. MODELO PARA DEFINIR MODALIDAD CLAVE  
(KEYFRAMING) CON ATRIBUTO DE OBJETO  
COMBINADO

NOTE FINAL  
12x12



### Figura 4

**BLENDED OBJECT ATTRIBUTE KEYFRAMING MODEL****FIELD OF THE INVENTION**

[0001] This invention relates in general to the field of computer graphics. More particularly, this invention relates to animation authoring tools.

**BACKGROUND OF THE INVENTION**

[0002] The industry standard for animation authoring tools uses a key frame, which is defined as a point in time, and a set of property changes that occur at that point in time. The properties can be anything from the color of an object to the entire contents of the scene, for example. Some tools, such as Macromedia Flash, represent key frames at the layer level and store the entire state of all objects in the layer at that point in time. Other tools, such as Adobe Live Motion, represent key frames on the attributes of an object, so that an indicator is stored on every property of an object which tells whether or not the property is animated. Both approaches have several drawbacks.

[0003] A drawback of representing key frames at the layer level and storing the entire state of all objects in the layer at that point in time is that it is difficult for the user to determine from the user interface which properties are being animated on any particular object. Another drawback of this approach is that it is very difficult for the user to animate the value of a property across one of these key frames. For example, if the user wants the value of the color of an object to change from red to blue starting at 1 second, and ending at 3 seconds, he could do this by setting up a key frame at 1 second containing the color red and a key frame at 3 seconds containing the color blue. If the user then wanted to start the object moving at time 2, he could do this by setting a key frame at time 2. However, because of the key frame model, this key frame would also contain the value of the color at time 2. If the user ever changed the time value of this key frame, the color animation would change as well. There is no way for the user to resolve this other than by manually resetting the color value stored in that key frame so that a smooth interpolation from red to blue is restored.

[0004] The approach of representing key frames only on the attributes of an object has a drawback of, that in order to animate any property on an object, the user must search through a list of all properties on the object, and set the switch that makes that property animated, allowing key frames to be stored for that property. Furthermore, this approach requires the user to select or click a button on each property that he desires to animate. This can be cumbersome when trying to author an animation. Moreover, using this approach it is difficult to quickly determine at what times particular properties of an object are being animated, if the user cannot see all of the properties for the element on screen.

[0005] Many of the 2d animation tools currently on the market (Flash, After Effects, etc.) have animation as their primary mode of authoring. That is, most of the time that the user is interacting with the tool, he is working on an animation. However, as a tool whose primary task is user interface design, it is more desirable to allow the user to interact with the scene without having to think about animation. For instance, a user may keyframe a wiggle animation on a button which he has placed on his form, and then decide that the button is not in the correct position on the form. It would be desirable for the user to be able to do this without modifying all of the key frames in the wiggle animation. In other 2d tools, the user has to be constantly aware of where the current time marker is in the timeline used in the designing or authoring (and in some tools whether or not the property being edited can be keyframed).

[0006] In view of the foregoing, there is a need for systems and methods that overcome the limitations and drawbacks of the prior art.

### **SUMMARY OF THE INVENTION**

[0007] The present invention is directed to combined object level and property level key framing systems, methods, and user interfaces. A user can enter a mode where any change to a property is recorded as a key framing action (e.g., creation or edit) at the current time as indicated by the timeline. When a key frame is created in this mode, the surrounding object level key frames are inspected to ensure that the functionality that users expect from object level key frames is maintained. If a user explicitly creates an

animation of an attribute that spans an object level key frame, then that span for the animation is maintained as the user edits.

**[0008]** Exemplary embodiments include a method of keyframing an object in which at least one property and a time for the object are identified. As set forth further below, e.g., with respect to Figures 1-3, an attribute key frame is a key frame at the attribute level and a compound key frame is a key frame at the object level. A compound key frame can be thought of as essentially a placeholder for a “virtual attribute key frame” on all possible attributes. Other attribute key frames force this virtual attribute key frame to manifest itself on those attributes, as set forth below.

**[0009]** A first compound key frame is created at a first time, and then a second time is received for the object, e.g., by moving a playhead in a timeline element. Another compound key frame is created at the second time. Additional times for the object may be received, followed by creation of associated compound key frames at each of the additional times. A change to the property or properties may be received prior to creating the second compound key frame, in which case the second compound key frame incorporates the changes. Additionally, an attribute key frame may be created responsive to the received property change(s) if no attribute key frame exists for the property or properties at the time the received change is received. Alternately, an existing attribute key frame may be changed responsive to the received property change(s) if the existing attribute key frame exists at the time the received change(s) is received.

**[0010]** According to additional embodiments, an object may be keyframed by receiving a value for an attribute for the object at a first time, and then, if an attribute key frame corresponding to the attribute exists at the first time, amending the attribute key frame responsive to the received value for the attribute; otherwise, populating a neighboring object key frame (also referred to herein as a compound key frame) with an attribute key frame if an object key frame exists.

**[0011]** Populating the neighboring object key frame may comprise (1) if the neighboring object key frame exists later in time than the first time, and an attribute key frame exists later in time than the first time, then setting a first value to the value of the attribute key frame that exists later in time than the first time; or (2) if the neighboring object key frame that exists later in time than the first time, and an attribute key frame

does not exist later in time than the first time, then setting the first value to the value of the originally received value for the attribute; and creating a new attribute key frame at the neighboring object key frame using the first value.

[0012] Additionally, populating the neighboring object key frame may comprise (1) if the object key that exists is earlier in time than the first time, and an attribute key frame exists earlier in time than the first time, then setting a first value to the value of the attribute key frame that exists earlier in time than the first time; or (2) if the object key that exists is earlier in time than the first time, and an attribute key frame does not exist earlier in time than the first time, then setting the first value to the value of the originally received value for the attribute; and creating a new attribute key frame at the neighboring object key frame using the first value.

[0013] Additional features and advantages of the invention will be made apparent from the following detailed description of illustrative embodiments that proceeds with reference to the accompanying drawings.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

[0014] The foregoing summary, as well as the following detailed description of preferred embodiments, is better understood when read in conjunction with the appended drawings. For the purpose of illustrating the invention, there is shown in the drawings exemplary constructions of the invention; however, the invention is not limited to the specific methods and instrumentalities disclosed. In the drawings:

[0015] Figure 1 is a diagram of an exemplary timeline element with playhead in accordance with the present invention;

[0016] Figure 2 is a diagram of an exemplary timeline element in accordance with the present invention;

[0017] Figure 3 is a diagram of another exemplary timeline element showing exemplary keyframing in accordance with the present invention;

[0018] Figure 4 is a flow chart of an exemplary method of keyframing in accordance with the present invention;

[0019] Figure 5 is a diagram of an exemplary object in a scene with user interface in accordance with the present invention;

[0020] Figures 6A-6F are diagrams of exemplary timeline elements useful for illustrating a method of creating animation elements in accordance with the present invention;

[0021] Figure 7 is a flow diagram of an exemplary process of creating animation elements in accordance with the present invention;

[0022] Figures 8A-8H are diagrams of exemplary timeline elements useful for illustrating exemplary auto-adding key frame scenarios in accordance with the present invention;

[0023] Figures 9A and 9B are diagrams of exemplary timeline elements useful for illustrating exemplary compound key frame scenarios in accordance with the present invention; and

[0024] Figure 10 is a block diagram showing an exemplary computing environment in which aspects of the invention may be implemented.

### **DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS**

[0025] The present invention is directed to combined object level and property level key framing systems, methods, and user interfaces. A exemplary key frame in accordance with the present invention comprises the geometry and property values of an object at a particular time, and can be created against an individual object. Such a key frame can be shown when an object is animated at a particular time. A user can enter a mode in which any change to a property (also referred to herein as an attribute) is recorded as a keyframing action, such as creation or edit, at the current time as indicated by, or on, a timeline element. When a key frame is created in this mode, the surrounding object level key frames are inspected to ensure that the functionality that users expect from object level key frames is maintained. If a user explicitly creates an animation of an attribute that spans an object level key frame, then that span for the animation is maintained as the user edits.

[0026] An exemplary key frame authoring user interface (UI) is shown in Figure 1, and comprises a compound key frame 10 and an attribute key frame 20. The exemplary key frame authoring UI timeline element 5 represents key frames at the object level (called compound key frames) and at the attribute level (called attribute key

frames), and leverages both models of key framing to make the creation of key framed animations more efficient. A playhead 7 is provided that can be moved to a particular time to set properties, key frames, etc.

[0027] When a new attribute key frame 20 is created, it is represented at the object level with compound (or object level) key frame 10 on the time line. Object level key frames can be created explicitly which, given the behavior for adding key frames described below, will effectively represent the entire state of the attributes on an object at that point in time without actually creating attribute key frames for each one of those attributes at that time. For instance, if the user selects an object in the scene, and then selects a "new key frame" button (or actuates or selects a substantially similar feature), then a new object level key frame is created at the current time. As new attribute key frames are added in the regions of time adjacent to that compound key frame, it is populated with attribute key frames to maintain the state of the object at that time.

[0028] As a user adds key frames to the animation, object level key frames are used to determine the span of newly animated attributes. Consider the scenario in which a user has authored an animation that moves an object from a position of (10,10) to a position of (100,100) from time 1 to time 2. Such an exemplary timeline element (which may be provided or displayed as part of a UI) is shown in Figure 2, in which the object "oval 1" is set to move from a first position A (assume A corresponds to underlying position (10,10)) to a second position B (assume position B corresponds to underlying position (100,100)). So at time 1, an object level key frame is generated for the beginning, and at time 2, an object level key frame is generated for the end.

[0029] Now assume the user wants the object to scale up to 150% of its original size, and back down to its original size over the course of the movement from time 1 to time 2. In accordance with the present invention, this can be accomplished by moving the current time marker (i.e., the playhead 7) to time 1.5, as shown in Figure 3, making sure that the timeline element is recording changes, and changing the value of the scale property to 150% at that point in time. In other words, at time 1.5, the scale is changed, and an object level snapshot is created. A population process, such as that described with respect to Figure 4, is then used to determine the property (attribute) key frames at the

current time (time 1.5) at the adjacent object levels (the ones directly previous and after (here, time 1 and time 2). Conventional interpolation algorithms may be used.

[0030] More particularly, when the user makes this change as shown in Figure 3, the compound key frame that exists before this newly created key frame (here, element 12 at time 1) is examined, along with the compound key frame immediately following the newly created key frame in time (here, element 16 at time 2). If either or both of these key frames do not have attribute key frames for the scale of the object, then the appropriate values for scale is filled in for those key frames. In this scenario, attribute key frames 22, 26, respectively, for a value of 100% scale would be created under the previous compound key frame 12, and underneath the next compound key frame 16, as shown in Figure 3.

[0031] Logic for the above described exemplary process is described in more detail with respect to the flow chart of Figure 4. In Figure 4, the neighboring object level key frames are inspected and populated. When a new key frame is added, a walk backward and forward is conducted. If an attribute key frame does not exist, nor does an object key frame exist (both earlier in time and later in time), then a new attribute key frame is created, which causes the creation of an object level key frame.

[0032] More particularly, at step 405, an attribute value is changed in the animation mode. It is determined at step 410 whether an attribute key frame exists at the current time. If so, then the key frame value is amended at step 490, and the process completes. If, however, an attribute key frame does not exist at the current time, it is determined whether an object key frame exists later than the current time, at step 415.

[0033] If an object key frame does not exist later in time at step 415, then it is determined at step 420 whether an object key frame exists earlier than the current time. If not, then a new attribute key frame is created for the current time at step 495, but if an object key frame does exist earlier than the current time, then it is determined at step 425 if there is an attribute key frame for this attribute earlier than the current time. If there is an attribute key frame for this attribute earlier than the current time, then the value at that key frame is maintained at that key frame, at step 455. If there is no attribute key frame for this attribute earlier than the current time, then the original value of the property is maintained as the value, at step 450. In either case, after step 450 or step 455, the earlier

object key frame is populated with the key frame containing the value of the held or maintained value from step 450 or step 455, at step 460. Processing then continues at step 495 with a new attribute key frame being created for the current time. It is noted that in the case when the current time is  $\text{time}=0$ , then there is no need to look earlier in time, because  $\text{time}=0$  is taken to be the smallest possible time value.

[0034] If, from step 415, it is determined that an object key frame exists later than the current time, then, at step 430, it is determined whether there is an attribute key frame for this attribute that exists later than the current time. If there is an attribute key frame for this attribute later than the current time, then the value at that key frame is maintained at that key frame, at step 440. If there is no attribute key frame for this attribute later than the current time, then the value used is the value of the last attribute key frame recorded or a base value (which may be predetermined) if none exists, at step 435. In either case, after step 435 or step 440, the later object key frame is populated with the key frame containing the value of the held or maintained value from step 435 or step 440, at step 445. Processing then continues at step 420, as set forth above.

[0035] The exemplary key frame population method described above with respect to Figure 4 occurs on the creation of a new key frame. When the current time marker is over an existing attribute key frame, and that attribute is modified while keyframing is enabled, then the value of the key frame at that time is modified. This allows the system to maintain time spans explicitly created by the user. For example, if the user explicitly views the attributes of an object and deletes an attribute key frame that is between two other attribute key frames, then that time span will be maintained if either of the attribute key frames bounding the time span is modified.

[0036] The exemplary method for recording the values at key frames allows the system to specify a mode of interaction where all edits made by the user to any property automatically generates animation key frames. This mode can be entered, for example, by a user pressing a toggle button which toggles the key frame mode. This mode makes the quick creation of a large number of key frames possible in a short period of time. The number of clicks or other input required by the user is significantly reduced over other keyframing interfaces.

[0037] When in the keyframing mode, it becomes desirable to indicate to the user that he is setting key frames with nearly every edit that he does. This is preferably accomplished by providing an indicator in context with the edit to notify the user that a key frame for the edit which he just performed has been recorded. This indicator also gives the user a button which can be selected to cancel the key frame that had just been set or to choose other options for interpreting the change just made. This on-object user interface is provided in the context of the change, so, with respect to Figure 5, if the user drags an object 510 in the scene 500 changing its position, the UI 520 will appear in the scene. For example, if the user changes the color of the object in the color picking UI, the on-object user interface is popped up over the color editor. An on-object UI indicates when changes done in the rest of the authoring tool UI are being recorded by the animation system. When a property is changed during keyframing, there is desirably a visual indicator that indicates that a key frame was changed. Items that can be changed include x, y, height, width, color, rotation, skew, for example.

[0038] There are two separate ways of interacting with objects in the scene, one which is animation focused, and one which is layout focused. In "animate" mode, the user is specifically focused on creating key frames for animations. Changes that the user makes to properties will be automatically recorded as key frames in an animation of those properties. In "layout" mode, the user's changes will be interpreted as changes to the non-animated values of properties where possible.

[0039] One exemplary way of segmenting the set of possible properties on an element is to break them into composing and non-composing properties. Composing properties are properties which can be meaningfully composed through simple operations. For example, a float is a composing property value because it can be composed with another value by addition. Non-composing properties cannot be meaningfully composed by simple operations. For example, a color value is not a composing value because composing it through an operation such as addition would cause out of gamut conditions, and other undesired behavior.

[0040] A reason to distinguish between these two types of properties is that, when they are animated, composing properties can be meaningfully changed in layout mode, while non-composing properties cannot. For example, if the user has an animation

on the top property of an element (whose value is a float) and then the user moves that element in layout mode, then that change can be meaningfully composed with the animation (the element will now animate from its current position). More specifically, the animation that the user specified on the top property is relative to the current top value on the element. This is possible because the top property has a value that is composing (the animation engine takes the original position of the element, and adds the value output by the animation to get the current, animated position of the element). Thus, when the user moves the element in layout mode, the base value for the top property is updated, and the animation engine will automatically pick up the change, and animate from the new position of the element.

[0041] When the user changes an animated, non-composing property in layout mode, the system desirably indicates that that change cannot be meaningfully incorporated with the property. For example, either the user forgot that he is in layout mode, and meant to create a key frame at the current time, or the user had something else in mind such as changing the base value of that property. In this case, the user will be preferably warned that the new value cannot be incorporated into the current animation without creating a key frame. Additionally, a marker is desirably added to the timeline to indicate where this action will happen, and the value will desirably be visible in the scene. If the user enters animate mode, then the values will be converted into key frame values. If the user seeks the timeline, then the new values that he added will be lost, as they will be overridden by the animation.

[0042] A compound key frame is meant to represent the state of the object to which it belongs at the time where it is set on the timeline. However, in order to maintain an efficient representation of animations, only the changes to properties that affect animation are desirably stored, and represented in the user interface.

[0043] In animate mode, a user may select, for example, to add a new key frame, add a shape, change a property, or change a selection. If a new key frame is to be added, then the process set forth above with respect to Figure 4 may be implemented, and further examples are given below (e.g., with respect to Figure 9). If the user desires to add a shape at a time later than 0, then the base value for opacity is preferably set to 0, and a key frame is added at the current time, setting the opacity to 1.

17

[0044] If the user chooses to change a property, then it is determined whether a key frame exists for this property at the current time. If so, then the key frame value at the current time is modified accordingly. If a key frame does not exist for this property at the current time, then a new attribute key frame is created (see, e.g., Figure 8).

[0045] If the user chooses to change a selection, it is determined whether the current selection has compound key frames. If so, the selection is changed accordingly. If not, then a compound key frame is created at the current time.

[0046] According to aspects of an embodiment, in layout mode, the user may convert temporary key frames to actual key frames or generate temporary key frames. For example, if a user changes a property, then, if the property is composing, the base value for the property is desirably modified such that the property value meets the edited value at the current time. If the property is not composing, and if the property is animated, then the temporary key frames are shown. However, if the property is not composing, and is not animated, then the property value is desirably updated and the process returns to the start state for layout mode.

[0047] Alternatively to the temporary key frames, a user interface could provide a user with the option to edit base values (e.g., by selecting a “none” tab in the user interface), or set animation key frames (e.g., by selecting a timeline tab). In such an embodiment (when the user is on the “none” tab), the current time is not active (i.e., not set to a particular value), and animation mode is not active. It is noted that in a specific timeline’s tab, the current time and animation mode are both active.

[0048] An exemplary system begins in the layout mode. Before any animations are present, the user is likely going to arrange controls on the form as desired for the static layout of his UI. When the user presses the animate button, he desirably enters animate mode where key frames are automatically created for values that he changes. In this mode, the add new key frame button is desirably available.

[0049] An example of animation creation is now described, with respect to Figures 6A – 6F and Figure 7. Assume the user wishes to create a simple animation that moves an oval from left to right over half a second, holds the oval still for half a second, and then moves the oval back to its original position. Initially, the oval is created (step 700) and placed at a position on the screen and given a start time (step 705). The animate

mode is entered (step 710) by clicking on an animate button or making a similar type selection. A new compound key frame is created for the oval at the current time (step 715). For example, the start time for the animation is identified on the timeline at time = 0.5, in Figure 6A.

[0050] The playhead 600 is moved forward 0.5 seconds, as shown in Figure 6B, to 1 second (step 720). Then, as shown in Figure 6C, the position (e.g., X,Y position) of the oval is changed, and key frames are added at 1 second (step 725). Processing continued at step 720. The playhead 600 is moved forward another 0.5 seconds, from 1 second to 1.5 seconds, as shown in Figure 6D, and a key frame is added (e.g., by a user activating an “add keyframe” button for example, on the UI), as shown in Figure 6E, to specify that the oval should keep its position between 1 and 1.5 seconds.

[0051] The playhead 600 is then moved forward another 0.5 seconds, to time = 2, at the position of the oval is changed back to its starting point and a key frame is added accordingly, as shown in Figure 6F. At this point, the user has created the desired animation, and can exit animate mode to change the layout of the scene without creating key frames, or stay in animate mode to keyframe additional animations.

[0052] If a user is in animate mode, the user may add and adjust the key frames at the current point in time as he adjusts the objects attributes using the object handles and palettes. For each attribute the user changes at a given time, an attribute key frame is added. Compound key frames on the object are also taken into account to preserve their meaning. Moreover, desirably, when the user changes a property and there is an attribute key frame for that property at the current time, then that attribute key frame is edited. Otherwise, when the user changes a property there is not an attribute key frame for that property at the current time, there are several cases as set forth below, examples of which are described with respect to Figures 8A-8H.

[0053] If the playhead 600 is on the only compound key frame on the element, as shown in Figure 8A, then an attribute key frame is added for the new value at the current time, as shown in Figure 8B.

[0054] If the playhead 600 is before all compound key frames on the element, as shown in Figure 8C, then, in this case, if the next compound key frame does not have an attribute key frame for the changed property, then it is populated with the old value of the

property. An attribute key frame is also created at current time with the new value, as shown in Figure 8D.

[0055] If the playhead 600 is disposed after all compound key frames on the element, as shown in Figure 8E, then, in this case, if the previous compound key frame does not have an attribute key frame for the changed property, then it is populated with the old value of the property. An attribute key frame is also created at current time with the new value, as shown in Figure 8F.

[0056] If the playhead 600 is disposed between two compound key frames on the element, as shown in Figure 8G, then in this case, both the compound key frame before current time, and the compound key frame after current time are checked. If either does not have an attribute key frame for the changed property, then it is populated with the old value of the property. An attribute key frame is also created at current time with the new value, as shown in Figure 8H. It is noted that the base value will not necessarily be used if there are attribute key frames specified at any time beyond the compound key frames (e.g., similar to that set forth above with respect to step 435 in Figure 4).

[0057] An exemplary pre-condition to the examples described with respect to Figures 8A-8H is that the auto key frame button is desirably activated. A desirable post-condition is that changes that the user makes to attributes are recorded as attribute key frames. Parameters of the examples described with respect to Figures 8A-8H are as follows. Times and values of attributes are desirably set while in auto-keyframing mode. If there was not a compound key frame at the current time, then one is desirably created and provided to parents of this element to represent the new attribute key frame. If properties were not previously visible, then they are provided for the user to see. It is contemplated that this exemplary parameter can be disabled as an option.

[0058] It is desirable that a compound key frame can be added to an element at the current time to preserve state while creating an animation. Hold key frames can be created with this functionality. It is contemplated that an exemplary UI would comprise an add key frame button that is enabled when in keyframing/animate mode. A user would desirably select a "add compound key frame" button or selector and a new compound key frame is added and populated at the current time. If there is already a compound key frame at current time, then no change would be made.

[0059] To populate a compound key frame, for each property that is currently animated on the element to which the compound key frame is being added (as shown in Figure 9A), the system desirably adds an attribute key frame comprising the current, animated value of the property. The new exemplary compound key frame is shown in Figure 9B. The new compound key frame is desirably populated with the current values of all animations at current time. Note that the user did not have to change a value to create this compound key frame.

[0060] An exemplary pre-condition to the compound key frame creation, is that the user seeks to the time at which he wants to set the key frame. For example, the user may select "Timeline->Add Key frame" on the UI or press the add key frame button on the timeline element.

[0061] A user may explicitly add attribute key frames. This may be desirable if the user would like a key frame with the same value as the current key frame. In such a case, in an exemplary UI, the user would scroll the current time to the desired time for the key frame, move the mouse over the timeline area for the desired time, and press the set key frame button. Alternatively, for example, the user may be provided with an option to select "add key frame" in a pull-down menu. A new attribute key frame is thus added to the attribute at the current time.

[0062] Thus, it is contemplated that an exemplary UI can maintain user preferences for attribute key frame times across edits to attribute key frame values. Moreover, exemplary user interfaces allow for the creation of both object level and attribute level key frames, populate attribute key frame values based on surrounding object key frames, and do not require the explicit creation of key frames for a particular object or attribute.

[0063] It is further contemplated that the present invention can be used with two dimensional transformations as well as three dimensional transformations, but should not be limited to just these types of transformations.

#### **Exemplary Computing Environment**

[0064] Figure 10 illustrates an example of a suitable computing system environment 800 in which the invention may be implemented. The computing system environment 800 is only one example of a suitable computing environment and is not

intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing environment 800 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 800.

[0065] The invention is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

[0066] The invention may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network or other data transmission medium. In a distributed computing environment, program modules and other data may be located in both local and remote computer storage media including memory storage devices.

[0067] With reference to Figure 10, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer 810. Components of computer 810 may include, but are not limited to, a processing unit 820, a system memory 830, and a system bus 821 that couples various system components including the system memory to the processing unit 820. The system bus 821 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video



Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus (also known as Mezzanine bus).

**[0068]** Computer 810 typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer 810 and includes both volatile and non-volatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer 810. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

**[0069]** The system memory 830 includes computer storage media in the form of volatile and/or non-volatile memory such as ROM 831 and RAM 832. A basic input/output system 833 (BIOS), containing the basic routines that help to transfer information between elements within computer 810, such as during start-up, is typically stored in ROM 831. RAM 832 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 820. By way of example, and not limitation, Figure 10 illustrates operating system 834, application programs 835, other program modules 836, and program data 837.

[0070] The computer 810 may also include other removable/non-removable, volatile/non-volatile computer storage media. By way of example only, Figure 10 illustrates a hard disk drive 840 that reads from or writes to non-removable, non-volatile magnetic media, a magnetic disk drive 851 that reads from or writes to a removable, non-volatile magnetic disk 852, and an optical disk drive 855 that reads from or writes to a removable, non-volatile optical disk 856, such as a CD-ROM or other optical media. Other removable/non-removable, volatile/non-volatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 841 is typically connected to the system bus 821 through a non-removable memory interface such as interface 840, and magnetic disk drive 851 and optical disk drive 855 are typically connected to the system bus 821 by a removable memory interface, such as interface 850.

[0071] The drives and their associated computer storage media provide storage of computer readable instructions, data structures, program modules and other data for the computer 810. In Figure 10, for example, hard disk drive 841 is illustrated as storing operating system 844, application programs 845, other program modules 846, and program data 847. Note that these components can either be the same as or different from operating system 834, application programs 835, other program modules 836, and program data 837. Operating system 844, application programs 845, other program modules 846, and program data 847 are given different numbers here to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer 810 through input devices such as a keyboard 862 and pointing device 861, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 820 through a user input interface 860 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 891 or other type of display device is also connected to the system bus 821 via an interface, such as a video interface 890. In addition to the monitor, computers

may also include other peripheral output devices such as speakers 897 and printer 896, which may be connected through an output peripheral interface 895.

[0072] The computer 810 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 880. The remote computer 880 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 810, although only a memory storage device 881 has been illustrated in Figure 10. The logical connections depicted include a LAN 871 and a WAN 873, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the internet.

[0073] When used in a LAN networking environment, the computer 810 is connected to the LAN 871 through a network interface or adapter 870. When used in a WAN networking environment, the computer 810 typically includes a modem 872 or other means for establishing communications over the WAN 873, such as the internet. The modem 872, which may be internal or external, may be connected to the system bus 821 via the user input interface 860, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 810, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, Figure 10 illustrates remote application programs 885 as residing on memory device 881. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0074] As mentioned above, while exemplary embodiments of the present invention have been described in connection with various computing devices, the underlying concepts may be applied to any computing device or system.

[0075] The various techniques described herein may be implemented in connection with hardware or software or, where appropriate, with a combination of both. Thus, the methods and apparatus of the present invention, or certain aspects or portions thereof, may take the form of program code (i.e., instructions) embodied in tangible media, such as floppy diskettes, CD-ROMs, hard drives, or any other machine-readable storage medium, wherein, when the program code is loaded into and executed by a

25

machine, such as a computer, the machine becomes an apparatus for practicing the invention. In the case of program code execution on programmable computers, the computing device will generally include a processor, a storage medium readable by the processor (including volatile and non-volatile memory and/or storage elements), at least one input device, and at least one output device. The program(s) can be implemented in assembly or machine language, if desired. In any case, the language may be a compiled or interpreted language, and combined with hardware implementations.

[0076] The methods and apparatus of the present invention may also be practiced via communications embodied in the form of program code that is transmitted over some transmission medium, such as over electrical wiring or cabling, through fiber optics, or via any other form of transmission, wherein, when the program code is received and loaded into and executed by a machine, such as an EPROM, a gate array, a programmable logic device (PLD), a client computer, or the like, the machine becomes an apparatus for practicing the invention. When implemented on a general-purpose processor, the program code combines with the processor to provide a unique apparatus that operates to invoke the functionality of the present invention. Additionally, any storage techniques used in connection with the present invention may invariably be a combination of hardware and software.

[0077] While the present invention has been described in connection with the preferred embodiments of the various figures, it is to be understood that other similar embodiments may be used or modifications and additions may be made to the described embodiments for performing the same function of the present invention without deviating therefrom. Therefore, the present invention should not be limited to any single embodiment, but rather should be construed in breadth and scope in accordance with the appended claims.

26

**WHAT IS CLAIMED IS:**

1. A method of keyframing an object, comprising:
  - identifying at least one property and a time for the object;
  - creating a first compound key frame at the time;
  - receiving a second time for the object; and
  - creating a second compound key frame at the second time.
2. The method of claim 1, further comprising receiving additional times for the object and creating associated compound key frames at each of the additional times.
3. The method of claim 1, wherein receiving the second time for the object comprises moving a playhead to a position on a timeline in a user interface, the position corresponding to the second time.
4. The method of claim 1, further comprising entering an animate mode prior to creating the first compound key frame.
5. The method of claim 1, wherein each of the first and second compound key frames represents the state of the at least one property on the object at the associated time.
6. The method of claim 1, further comprising receiving a change to the at least one property prior to creating the second compound key frame, the second compound key frame incorporating the change to the at least one property.
7. The method of claim 6, further comprising creating an attribute key frame responsive to the received change to the at least one property if no attribute key frame exists for the at least one property at the time the received change is received, and changing an existing attribute key frame responsive to the received change to the at least one property if the existing attribute key frame exists at the time the received change is received.

8. A method of keyframing an object, comprising:

receiving a value for an attribute for the object at a first time;

if an attribute key frame corresponding to the attribute exists at the first time, then amending the attribute key frame responsive to the received value for the attribute;

otherwise, populating a neighboring object key frame with an attribute key frame if an object key frame exists.

9. The method of claim 8, further comprising creating an attribute key frame at the first time if neither an attribute key frame nor an object key frame exists at the first time.

10. The method of claim 8, wherein populating the neighboring object key frame comprises:

if the neighboring object key frame exists later in time than the first time, and an attribute key frame exists later in time than the first time, then setting a first value to the value of the attribute key frame that exists later in time than the first time;

if the neighboring object key frame that exists later in time than the first time, and an attribute key frame does not exist later in time than the first time, then setting the first value to the value of the last attribute key frame recorded or a base value if none exists; and

creating a new attribute key frame at the neighboring object key frame using the first value.

11. The method of claim 10, further comprising:

if the object key that exists is earlier in time than the first time, and an attribute key frame exists earlier in time than the first time, then setting a second value to the value of the attribute key frame that exists earlier in time than the first time;

if the object key that exists is earlier in time than the first time, and an attribute key frame does not exist earlier in time than the first time, then setting the second value to the value of the originally received value for the attribute; and

creating a new attribute key frame at the neighboring object key frame using the second value.



12. The method of claim 8, wherein populating the neighboring object key frame comprises:

if the object key that exists is earlier in time than the first time, and an attribute key frame exists earlier in time than the first time, then setting a first value to the value of the attribute key frame that exists earlier in time than the first time;

if the object key that exists is earlier in time than the first time, and an attribute key frame does not exist earlier in time than the first time, then setting the first value to the value of the originally received value for the attribute; and

creating a new attribute key frame at the neighboring object key frame using the first value.

13. In a computer system having a graphical user interface including a display and a user interface selection device, a method of keyframing an object via a timeline element on the display, comprising:

receiving a selection signal indicative of the user interface selection device selecting at least one property and a time for the object;

displaying a first compound key frame at the time on the timeline element;

receiving a selection signal indicative of the user interface selection device selecting a second time for the object; and

displaying a second compound key frame at the second time on the timeline element.

14. The method of claim 13, further comprising receiving additional selection signals indicative of the user interface selection device selecting additional times for the object, and displaying associated compound key frames at each of the additional times on the timeline element.

15. The method of claim 13, wherein receiving the selection signal indicative of the user interface selection device selecting a second time for the object comprises receiving an

execution signal indicative of a user moving a playhead to a position on a timeline in the timeline element, the position corresponding to the second time.

16. The method of claim 13, further comprising receiving an execution signal indicative of a user selecting an animate mode prior to displaying the first compound key frame.

17. The method of claim 13, further comprising receiving a selection signal indicative of the user interface selection device selecting a change to the at least one property prior to displaying the second compound key frame, the second compound key frame incorporating the change to the at least one property.

18. The method of claim 17, further comprising displaying an attribute key frame responsive to the received change to the at least one property on the timeline element if no attribute key frame exists for the at least one property at the time the received change is received, and changing an existing displayed attribute key frame responsive to the received change to the at least one property if the existing displayed attribute key frame exists at the time the received change is received.

19. A display device having rendered thereon a timeline element for keyframing an object by receiving a selection signal indicative of a selection of at least one property and a time for the object; displaying a first compound key frame at the time on the timeline element; receiving a selection signal indicative a selection of a second time for the object; and displaying a second compound key frame at the second time on the timeline element.

20. The display device of claim 19, wherein the timeline element is adapted to receive additional selection signals indicative of a selection of additional times for the object, and display associated compound key frames at each of the additional times on the timeline element.

21. The display device of claim 19, wherein the timeline element comprises a movable playhead.

22. The display device of claim 19, wherein the timeline element is adapted to receive an execution signal indicative of a user selecting an animate mode.

23. The display device of claim 19, wherein the timeline element is adapted to receive a selection signal indicative of a selection of a change to the at least one property prior to displaying the second compound key frame, the second compound key frame incorporating the change to the at least one property.

24. The display device of claim 23, wherein the timeline element is further adapted to display an attribute key frame responsive to the received change to the at least one property on the timeline element if no attribute key frame exists for the at least one property at the time the received change is received, and to change an existing displayed attribute key frame responsive to the received change to the at least one property if the existing displayed attribute key frame exists at the time the received change is received.

25. A method in a computer system for displaying on a display device a keyframing of an object, the method comprising:

- receiving an indication from a user of a value for an attribute for the object at a first time;

- if an attribute key frame corresponding to the attribute, in accordance with the received user indication, exists at the first time, then displaying on the display device the attribute key frame responsive to the received value for the attribute;

- otherwise, displaying on the display device a neighboring object key frame with an attribute key frame if an object key frame exists.

26. The method of claim 25, further comprising displaying an attribute key frame at the first time if neither an attribute key frame nor an object key frame exists at the first time.

27. The method of claim 25, wherein displaying on the display device the neighboring object key frame comprises:

if the neighboring object key frame exists later in time than the first time, and an attribute key frame exists later in time than the first time, then setting a first value to the value of the attribute key frame that exists later in time than the first time;

if the neighboring object key frame that exists later in time than the first time, and an attribute key frame does not exist later in time than the first time, then setting the first value to the value of the last attribute key frame recorded or a base value if none exists; and

displaying a new attribute key frame at the neighboring object key frame using the first value.

28. The method of claim 27, further comprising:

if the object key that exists is earlier in time than the first time, and an attribute key frame exists earlier in time than the first time, then setting a second value to the value of the attribute key frame that exists earlier in time than the first time;

if the object key that exists is earlier in time than the first time, and an attribute key frame does not exist earlier in time than the first time, then setting the second value to the value of the originally received value for the attribute; and

displaying a new attribute key frame at the neighboring object key frame using the second value.

29. The method of claim 25, wherein displaying on the display device the neighboring object key frame comprises:

if the object key that exists is earlier in time than the first time, and an attribute key frame exists earlier in time than the first time, then setting a first value to the value of the attribute key frame that exists earlier in time than the first time;

if the object key that exists is earlier in time than the first time, and an attribute key frame does not exist earlier in time than the first time, then setting the first value to the value of the originally received value for the attribute; and

displaying a new attribute key frame at the neighboring object key frame using the first value.

32

30. In a computer system having a graphical user interface including a display and a user interface selection device, a method of providing an on-object user interface on the display, comprising:

- receiving a selection signal indicative of the user interface selection device
- selecting an object;

- receiving a selection signal indicative of the user interface selection device
- selecting a property of the object to change;

- displaying a user interface in proximity to the object.

31. The method of claim 30, wherein the displayed user interface indicates when changes done in an authoring tool user interface are being recorded by an animation system.

**ABSTRACT OF THE DISCLOSURE**

Combined object level and property level key framing systems, methods, and user interfaces are provided in which a user can enter a mode where any change to a property is recorded as a key framing action (e.g., creation or edit) at the current time as indicated by a timeline. When a key frame is created in this mode, the surrounding object level key frames are inspected to ensure that the functionality that users expect from object level key frames is maintained. If a user explicitly creates an animation of an attribute that spans an object level key frame, then that span for the animation is maintained as the user edits.

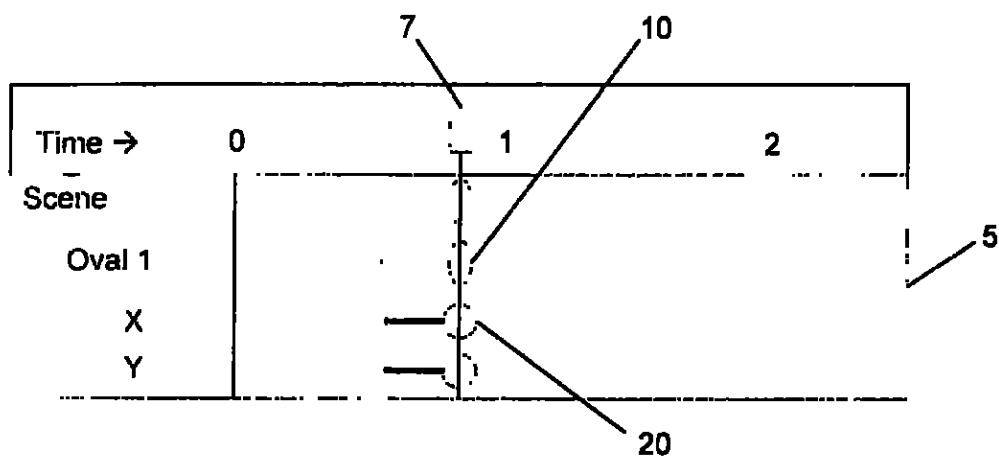


Fig. 1

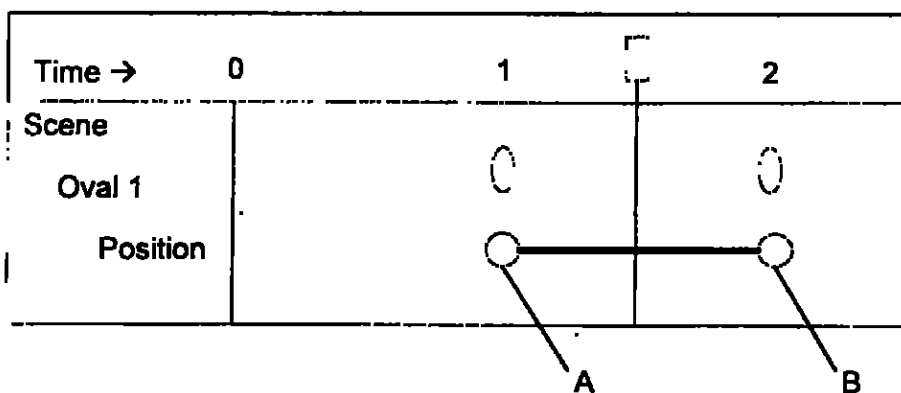


Fig. 2

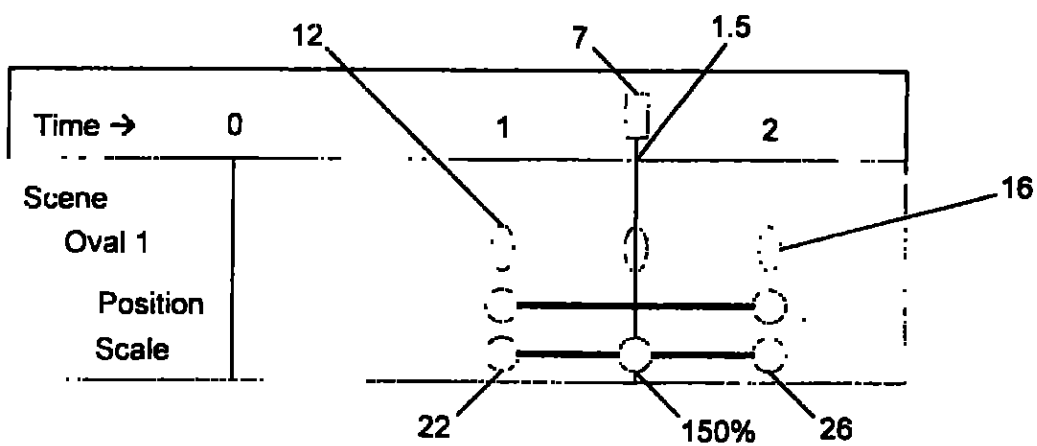
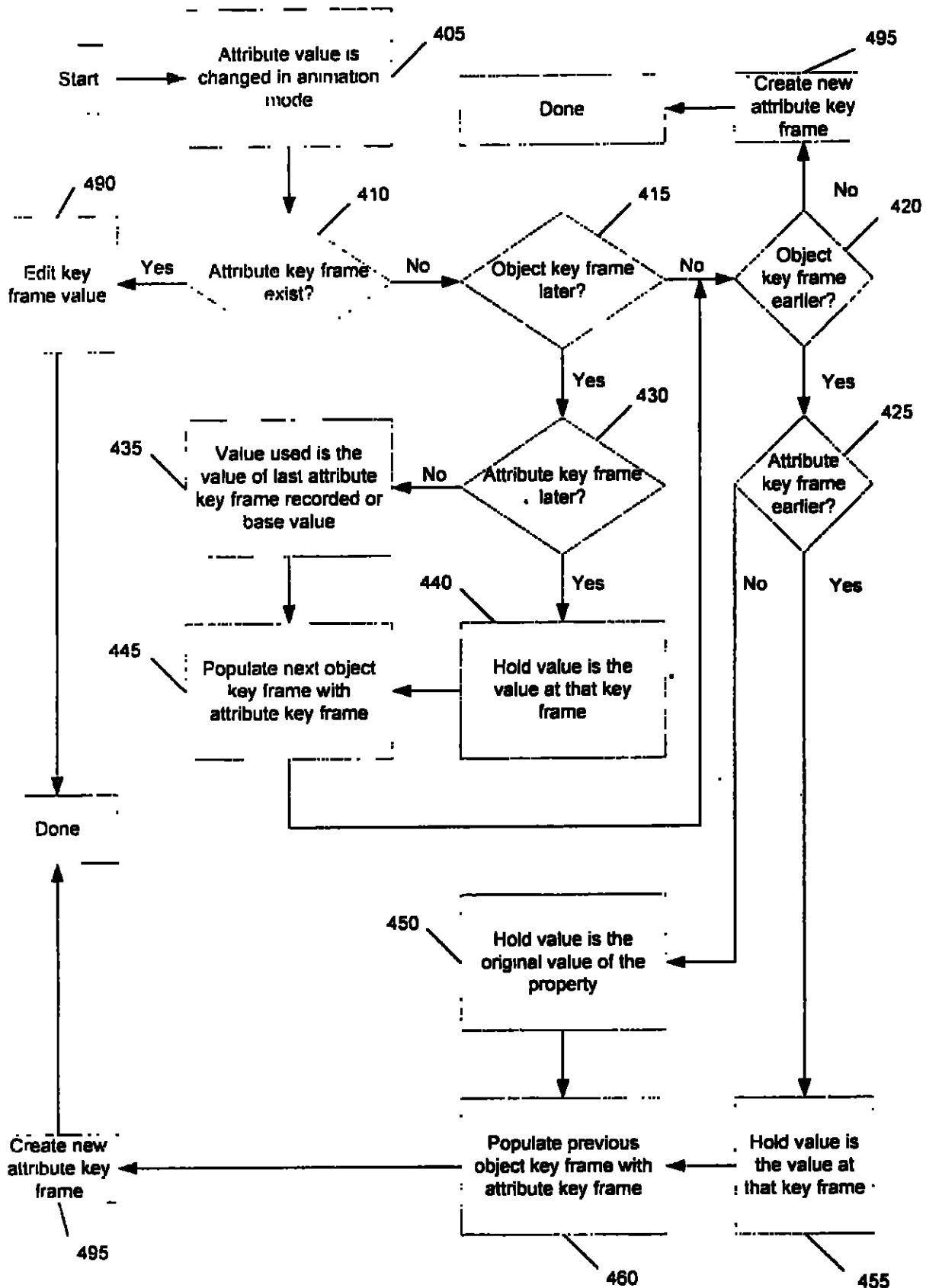
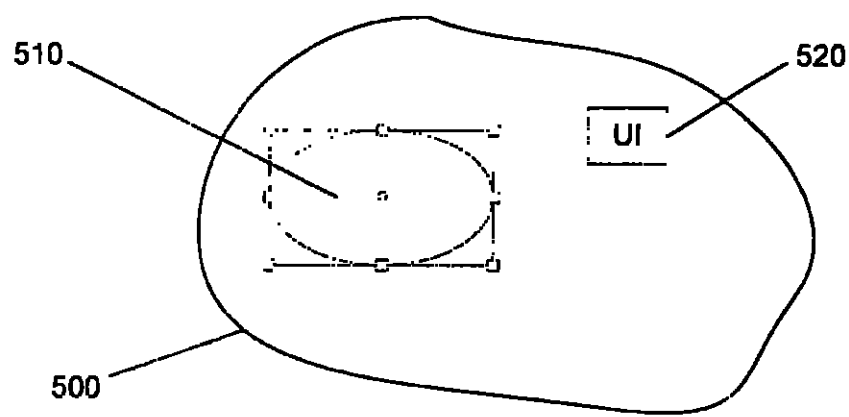


Fig. 3



**Fig. 4**



**Fig. 5**

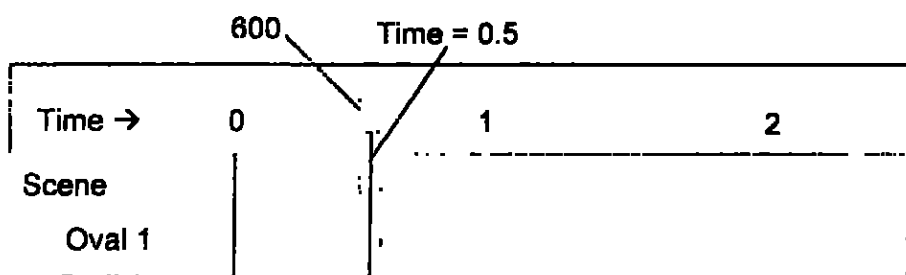


Fig. 6A

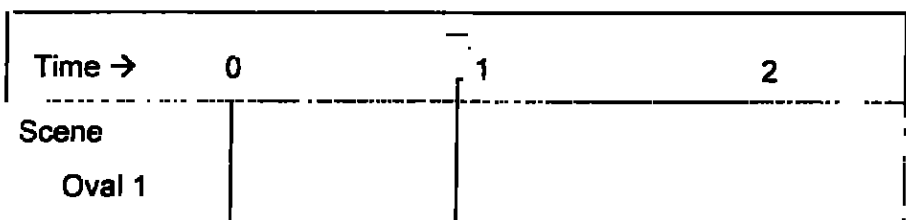


Fig. 6B

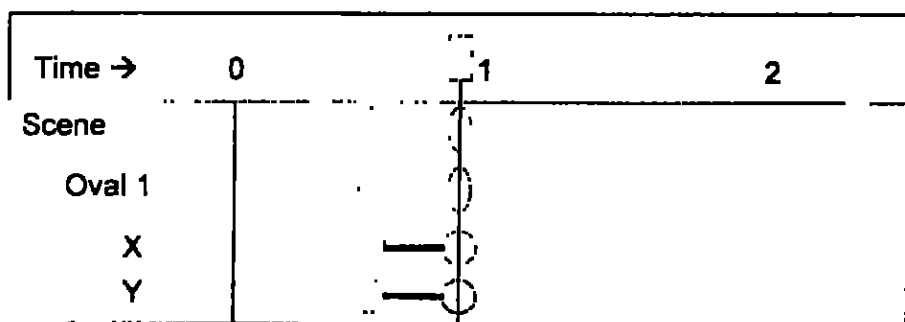


Fig. 6C

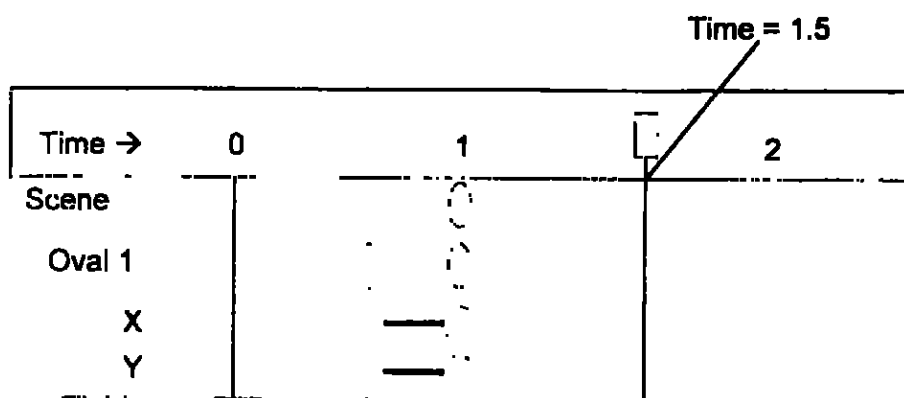


Fig. 6D

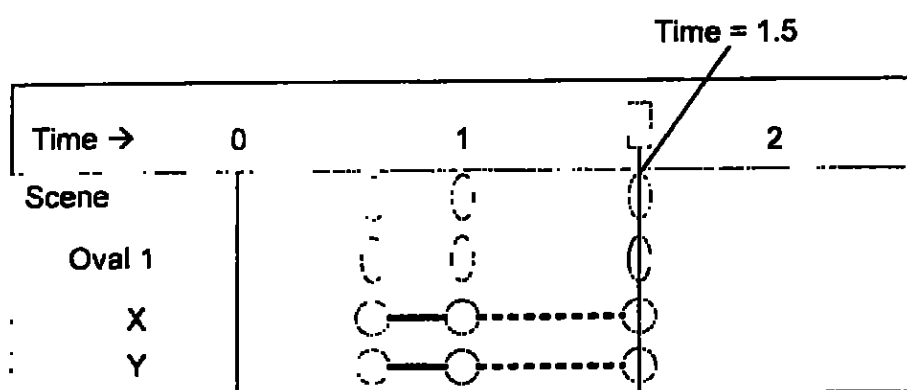


Fig. 6E

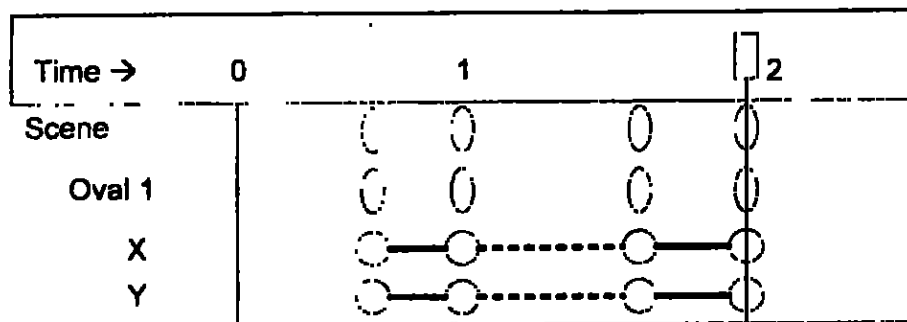
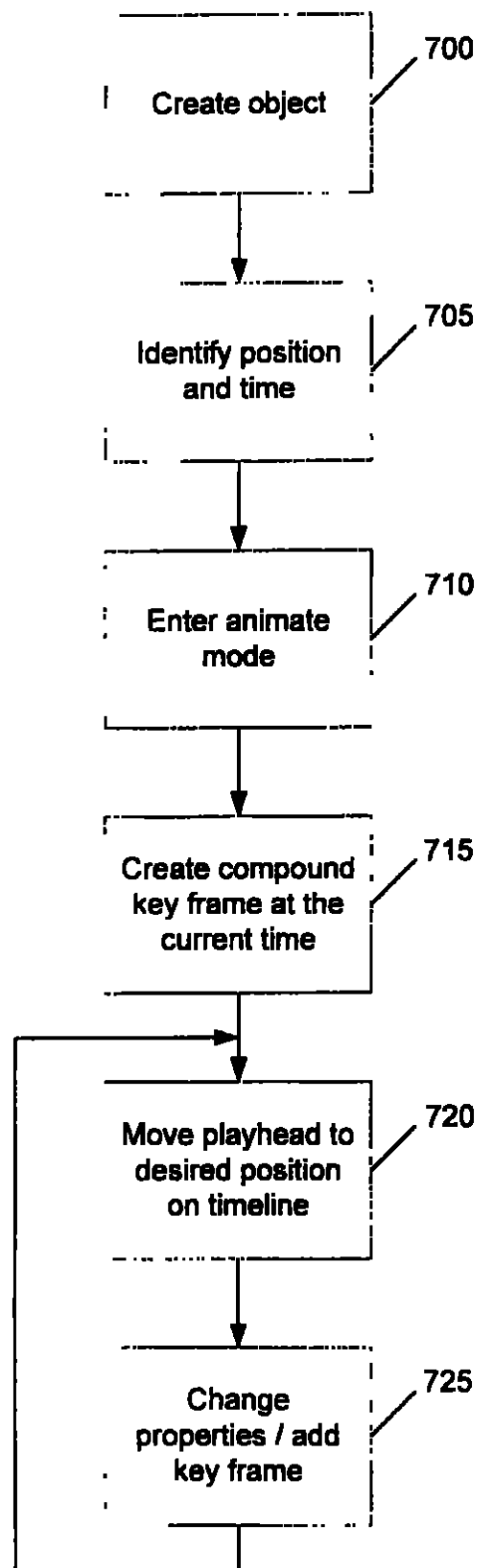


Fig. 6F



**Fig. 7**

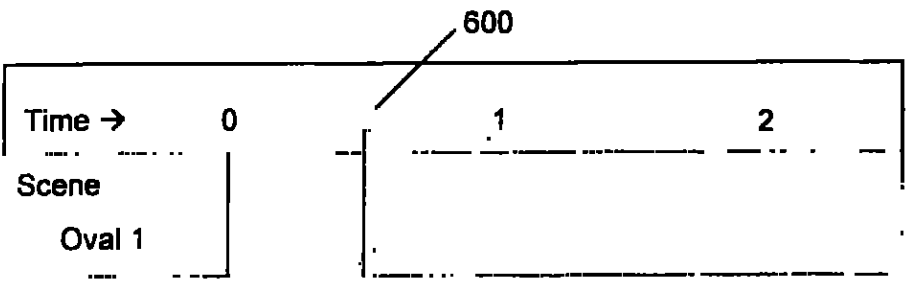


Fig. 8A

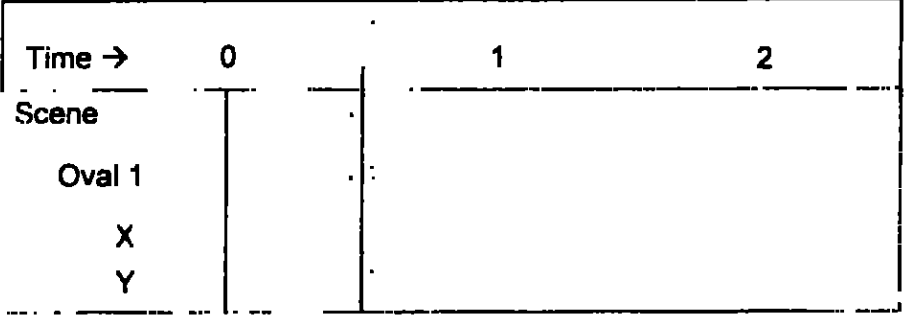


Fig. 8B

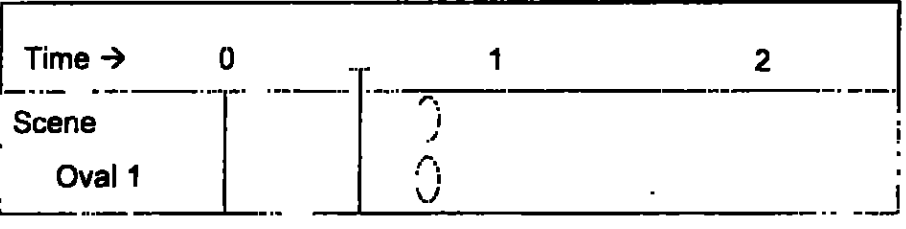


Fig. 8C

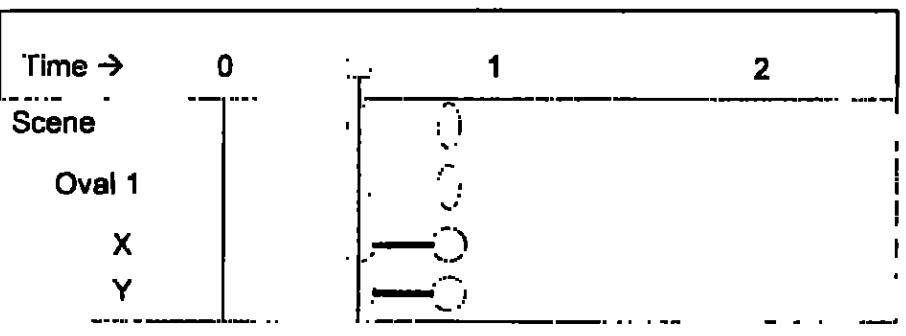


Fig. 8D

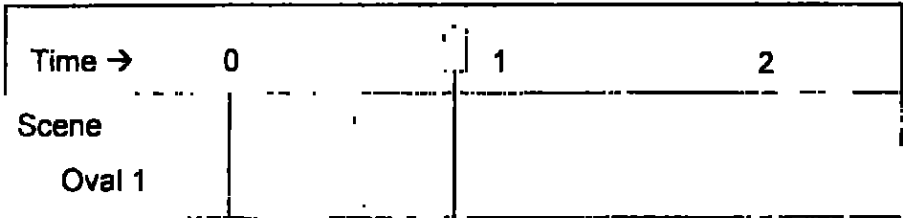


Fig. 8E

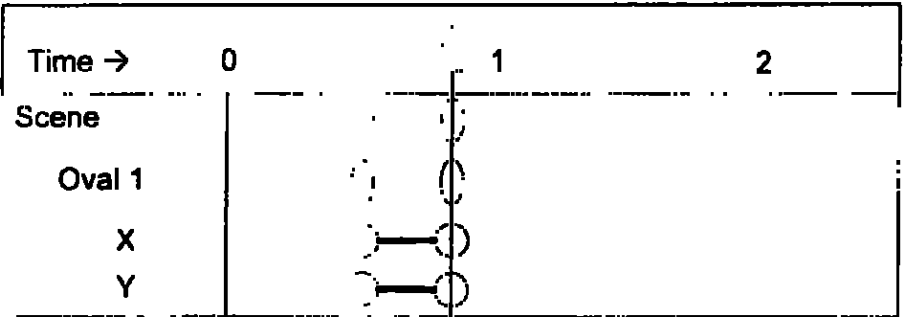


Fig. 8F

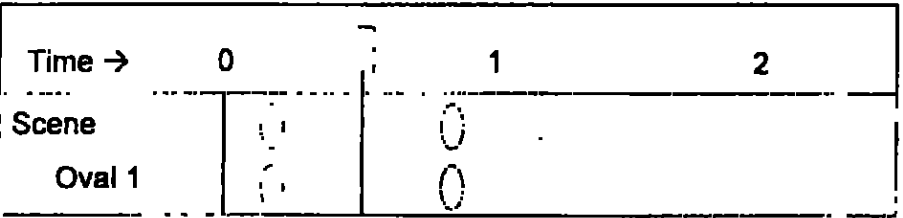


Fig. 8G

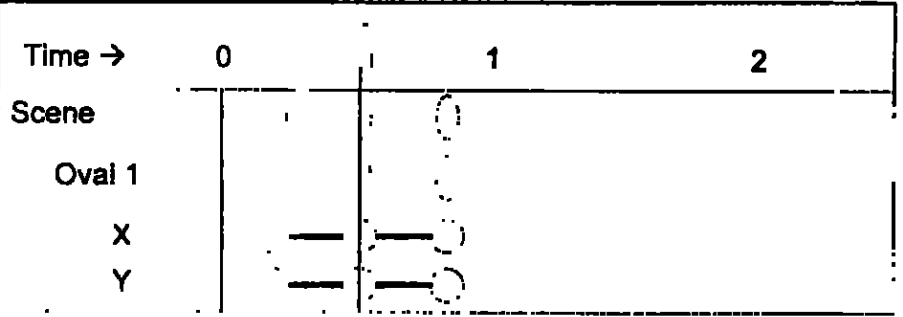


Fig. 8H

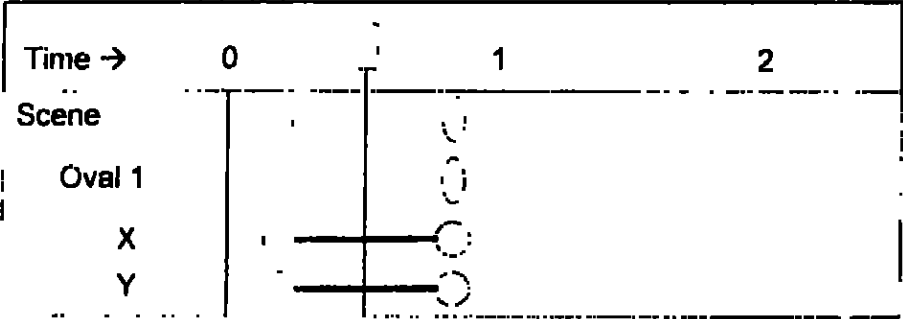


Fig. 9A

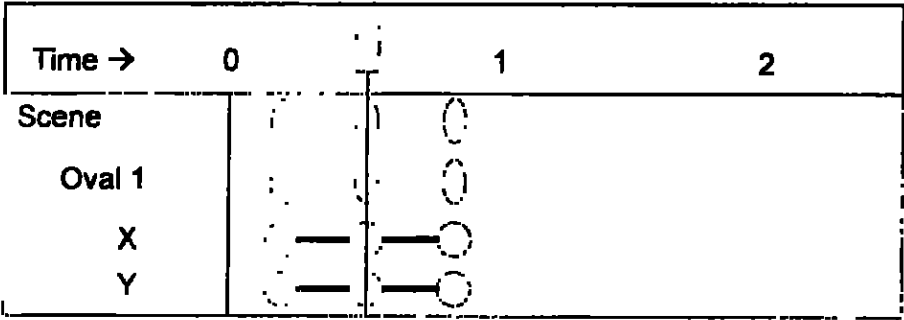
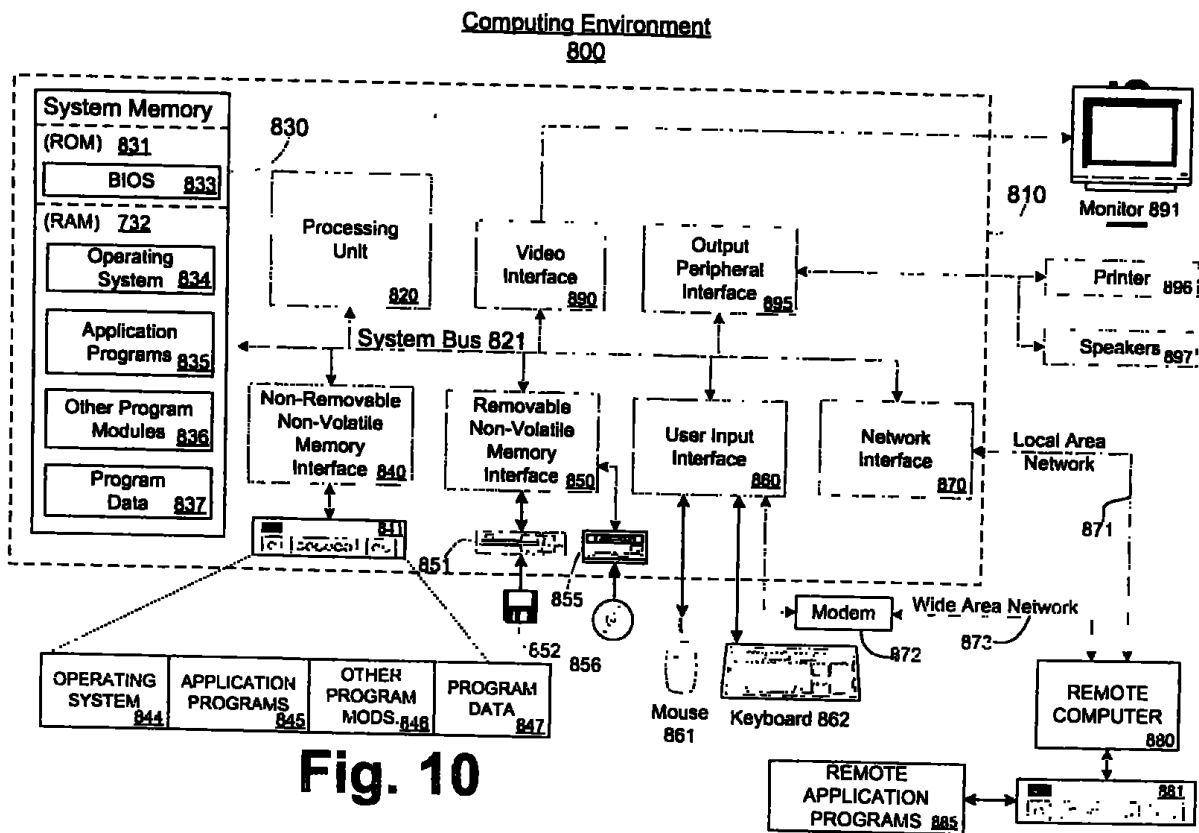


Fig. 9B



**Fig. 10**

44

**MODELO PARA DEFINIR MARCOS CLAVE (KEYFRAMING) CON  
ATRIBUTO DE OBJETO COMBINADO**

**CAMPO DEL INVENTO**

[0001] Este invento se relaciona en general con el campo de material gráfico en computadores. En particular, este invento se relaciona con herramientas de autoría de animación.

**ANTECEDENTES DEL INVENTO**

[0002] El estándar de la industria de herramientas de autoría de animación utiliza un marco clave, el cual se define como un punto en el tiempo y como un conjunto de cambios de características que ocurren en ese punto en el tiempo. Las características pueden ser cualquier cosa desde el color de un objeto hasta por ejemplo, el contenido total de la escena. Algunas herramientas, tal como la Macromedia Flash, representan los marcos clave en el nivel de estrato y almacenan el estado total de todos los objetos en el estrato, en ese punto en el tiempo. Otras herramientas, como el Adobe Live Motion, representan los marcos clave en los atributos de un objeto, de tal manera que se almacena un indicador en cada característica de un objeto, lo cual indica si la característica es animada o no lo es. Las dos aproximaciones tienen varios inconvenientes.

[0003] Un inconveniente de representar marcos clave en el nivel del estrato y de almacenar el estado completo de todos los objetos en el estrato en ese punto en el tiempo, consiste en que para el usuario es difícil determinar desde la interfaz de usuario cuales son las características que se están animando sobre cualquier objeto en particular. Otra desventaja de esta aproximación es que para el usuario es muy difícil animar el valor de una característica a lo largo de uno de éstos marcos clave. Por ejemplo, si el usuario quiere que el valor del color de un objeto cambie de rojo a azul, y que el cambio comience en 1 segundo y termine en 3 segundos, lo podría realizar estableciendo un marco clave que contenga el color rojo en 1 segundo y un marco clave que contenga el color azul a los 3 segundos. Si el usuario quisiera después que el objeto inicie el movimiento en el tiempo 2, podría hacerlo estableciendo un marco clave en el tiempo 2. Sin embargo, debido al modelo de marco clave, este marco clave también contendría el valor del color en el tiempo 2. Si el usuario llegare a cambiar el valor del tiempo de este marco clave, el color de la animación también cambiaría. Para el usuario no hay forma de resolver esto más que mediante el reestablecimiento manual del valor del color almacenado en ese marco clave, de tal forma que se reestablezca una interpolación fluida del rojo al azul.

[0004] La aproximación de representar los marcos clave únicamente en los atributos de un objeto tiene un inconveniente, y consiste en lo siguiente: si el usuario quiere animar una característica de un objeto, debe buscar a través de una lista de todas las propiedades en el objeto y graduar el selector que hace que esa característica se anime, permitiendo que se almacenen los marcos clave para esa característica. Además, esta aproximación le exige al usuario seleccionar o hacer clic en un botón en cada característica que él desee animar. Esto puede ser engorroso cuando se intenta crear una animación. Por otra parte, el uso de esta aproximación dificulta la rápida determinación del momento en que se están animando las características particulares de un objeto, si el usuario no puede ver en la pantalla todas las características para el elemento.

[0005] La mayoría de las herramientas de animación de 2 dimensiones que están actualmente en el mercado (Flash, After Effects, etc.) tienen la animación como su modo primario de autoría. Esto significa que la mayor parte del tiempo en que el usuario interactúa con la herramienta está trabajando en una animación. Sin embargo, al ser una herramienta cuya principal tarea es el diseño de interfaz de usuario, es más deseable que le permita al usuario la interacción con la escena sin que tenga que pensar en animación. Por ejemplo, un usuario puede definir un marco clave de animación con movimiento en un botón que ha ubicado en su modelo, y después decidir que el botón no está en la posición correcta en el modelo. Sería deseable para el usuario poder realizar esto sin tener que modificar todos los marcos clave en la animación con movimiento. En otras herramientas de 2 dimensiones, el usuario debe estar permanentemente al tanto de dónde se encuentra el indicador de tiempo en la línea de tiempos utilizada en el diseño o en la autoría (y en ciertas herramientas, debe estar al tanto de si a la característica que se está editando se le puede o no definir un marco clave).

[0006] En vista de lo antedicho, existe la necesidad de sistemas y de métodos que solucionen las limitaciones e inconvenientes de la técnica anterior.

### **RESUMEN DEL INVENTO**

[0007] El presente invento está dirigido a sistemas y métodos para definir marcos clave de nivel de objeto combinado y de nivel de características, así como a las interfaces de usuario. Un usuario puede ingresar a un modo en el que cualquier cambio a una característica se registra como una acción de definición de marcos clave (como por ejemplo creación o edición) en el tiempo actual, tal y como lo indica la línea de tiempos. Cuando se crea un marco clave en este modo, los marcos clave de nivel de objeto circundantes se inspeccionan para garantizar que se

mantenga la funcionalidad que los usuarios esperan de los marcos clave de nivel de objeto. Si un usuario crea explícitamente una animación de un atributo que se extiende sobre un marco clave de nivel de objeto, entonces a medida que el usuario edita se mantiene esa extensión para la animación.

[0008] Las incorporaciones de los ejemplos incluyen un método para definir marcos clave de un objeto en el cual se identifican al menos una característica y un tiempo para el objeto. Tal y como se establece más adelante, por ejemplo con relación a las Figuras 1-3, un marco clave de atributo es un marco clave en el nivel de atributo y un marco clave compuesto es un marco clave en el nivel de objeto. Un marco clave compuesto puede ser entendido esencialmente como un marcador de campo para un "marco clave de atributo virtual" en todos los atributos posibles. Otros marcos clave de atributo fuerzan este marco clave de atributo virtual para manifestarse él mismo en esos atributos, tal y como se establece más adelante.

[0009] En un primer tiempo se crea un primer marco clave compuesto, y después se recibe un segundo tiempo para el objeto, por ejemplo mediante el movimiento de una cabeza reproductora (*playhead*) en un elemento de línea de tiempos. Se crea otro marco clave compuesto en el segundo tiempo. Se pueden recibir tiempos adicionales para el objeto, seguidos por la creación de marcos clave compuestos asociados en cada uno de los tiempos adicionales. Se puede recibir un cambio en la(s) característica(s) antes de crear el segundo marco clave compuesto, caso en el que el segundo marco clave compuesto incorpora los cambios. Adicionalmente, se puede crear un marco clave de atributo sensible al (a los) cambio(s) de característica(s) recibido(s), en caso de que no exista un marco clave de atributo para la(s) característica(s), en el tiempo en que se recibe el cambio recibido. De forma alterna, se puede cambiar un marco clave de atributo existente sensible al (a los) cambio(s) de característica(s) recibido(s) si el marco clave de atributo existente existe en el tiempo en que se recibe(n) el(los) cambio(s) recibido(s).

[0010] De acuerdo con las incorporaciones adicionales, a un objeto se le pueden definir marcos clave mediante la recepción de un valor para un atributo para el objeto en el primer momento, y después, si un marco clave de atributo que corresponde al atributo existe en el primer momento, corrigiendo el marco clave del atributo sensible al valor recibido para el atributo; por lo demás, ocupar un marco clave del objeto cercano (también mencionado aquí como un marco clave compuesto) con un marco clave de atributo si existe un marco clave de objeto.

[0011] Ocupar el marco clave de objeto vecino puede incluir (1) si el marco clave del objeto cercano existe en el tiempo después del primer

momento, y un marco clave de atributo existe en el tiempo después del primer momento, entonces establecer un primer valor para el valor del marco clave del atributo que existe en el tiempo después del primer momento; o (2) si el marco clave del objeto cercano existe en el tiempo después del primer momento, y un marco clave de atributo no existe en el tiempo después del primer momento, entonces establecer el primer valor para el valor correspondiente al valor recibido originalmente para el atributo; y crear un nuevo marco clave de atributo en el marco clave del objeto cercano utilizando el primer valor.

[0012] Adicionalmente, ocupar el marco clave del objeto cercano puede comprender (1) si la clave de objeto que existe es anterior en el tiempo al primer momento, y un marco clave de atributo existe en el tiempo antes del primer momento, entonces establecer un primer valor al valor del marco clave de atributo que existe en el tiempo antes del primer momento; o (2) si la clave de objeto que existe es anterior en el tiempo al primer momento, y un a marco clave de atributo no existe en el tiempo antes del primer momento, entonces establecer el primer valor al valor correspondiente al valor recibido originalmente para el atributo; y crear un nuevo marco clave de atributo en el marco clave del objeto cercano utilizando el primer valor.

[0013] Se harán evidentes características y ventajas adicionales del invento a partir de la descripción detallada que sigue de las incorporaciones ilustrativas que están a continuación, con referencia a los dibujos que se acompañan.

### **BREVE DESCRIPCIÓN DE LOS DIBUJOS**

[0014] El anterior resumen, así como la siguiente descripción detallada de las incorporaciones preferidas, se comprenden mejor cuando se leen en conjunción con los dibujos adjuntos. Con el propósito de ilustrar el invento, se muestran construcciones de ejemplo del invento en los dibujos; sin embargo, el invento no se limita a los métodos específicos y formas de aplicación reveladas. En los dibujos:

[0015] La figura 1 es un diagrama de un ejemplo de elemento de línea de tiempos con una cabeza reproductora (*playhead*) de acuerdo con el presente invento;

[0016] La figura 2 es un diagrama de un ejemplo de elemento de línea de tiempos de acuerdo con el presente invento;

[0017] La figura 3 es un diagrama de ejemplo de otro elemento de línea mostrando una definición de marcos clave de acuerdo con el presente invento;

[0018] La figura 4 es un diagrama de flujo de un ejemplo de un método de definición de marcos clave de acuerdo con el presente invento;

[0019] La figura 5 es un diagrama de un ejemplo de un objeto en una escena con la interfaz de usuario, de acuerdo con el presente invento;

[0020] Las figuras 6A-6F son diagramas de ejemplos de los elementos de línea de tiempos útiles para ilustrar un método de creación de elementos de animación de acuerdo con el presente invento;

[0021] La figura 7 es un diagrama de flujo de un ejemplo de un proceso de creación de elementos de animación de acuerdo con el presente invento;

[0022] Las figuras 8A-8H son diagramas de ejemplos de elementos de línea de tiempos útiles para ilustrar escenarios de un ejemplo de auto-adición de marco clave, de acuerdo con el presente invento;

[0023] Las figuras 9A y 9B son diagramas de ejemplos de elementos de línea de tiempos útiles para ilustrar escenarios de un ejemplo de marcos clave compuestos, de acuerdo con el presente invento;

[0024] La figura 10 es un diagrama en bloque que muestra un ejemplo de un medio de cómputo en el cual ciertos aspectos del invento se pueden implementar.

#### **DESCRIPCIÓN DETALLADA DE LAS INCORPORACIONES PREFERIDAS**

[0025] El presente invento está dirigido a sistemas combinados de definición de marcos clave de nivel de característica y nivel de objeto, a métodos y a interfaces de usuario. Un ejemplo de marco clave de acuerdo con el presente invento, comprende los valores de geometría y de característica de un objeto en tiempo determinado, y puede ser creado contra un objeto individual. Tal marco clave se puede mostrar cuando se anima un objeto en un momento particular. El usuario puede ingresar a un modo en el cual cualquier cambio a una característica (también referenciada aquí como atributo) se registra como una acción de definición de marcos clave (tal como creación o edición) en el momento actual tal como se indica por/en un elemento de línea de tiempo. Cuando se crea un marco clave de este modo, los marcos clave del nivel de objeto cercanos se revisan para garantizar que se mantiene la funcionalidad que el usuario espera de los marcos clave de nivel de objeto. Si un usuario crea explícitamente una animación de un atributo que se extiende sobre un marco clave de nivel de objeto, entonces esta extensión se mantiene para la animación mientras que el usuario edita.

[0026] Un ejemplo de interfaz de usuario (UI, por sus iniciales en inglés) de autoría de marco clave se muestra en la Figura 1, y comprende un marco clave compuesto 10 y un marco clave de atributo 20. Un ejemplo del elemento de línea de tiempos 5 de interfaz de usuario de autoría marco clave representa los marcos clave al nivel de objeto (llamados marcos clave compuestos) y al nivel de atributo (llamados marcos clave de atributos), y apalanca los dos modelos de definición de marcos clave para hacer más eficiente la creación de animaciones con definición de marcos clave. Se provee una cabeza reproductora (playhead) 7, la cual puede ser desplazada a un tiempo particular para establecer las características, los marcos clave, etc.

[0027] Cuando se crea un nuevo marco clave de atributo 20, se representa al nivel de objeto con un marco clave compuesto 10 (o nivel de objeto) en la línea de tiempos. Los marcos clave de nivel de objeto se pueden crear explícitamente, los cuales, dado el comportamiento descrito a continuación para marcos clave que se adicionan, efectivamente representarán el estado completo de los atributos en un objeto en ese punto en el tiempo, sin crear en realidad marcos clave de atributos para cada uno de esos atributos en ese momento. Por ejemplo, si el usuario selecciona un objeto en la escena, y después selecciona un botón de "marco clave nuevo" (o acciona o selecciona una característica sustancialmente similar), entonces se crea un nuevo marco clave de nivel de objeto en el tiempo actual. A medida que se adicionan nuevos marcos clave de atributo en las regiones de tiempo adyacentes a ese marco clave compuesto, se ocupa con marcos clave de atributo para mantener el estado del objeto en ese tiempo.

[0028] A medida que un usuario adiciona marcos clave a la animación, los marcos clave de nivel de objeto se usan para determinar la extensión de los atributos recién animados. Considere el escenario en el que un usuario es el autor de una animación que mueve un objeto de una posición (10,10) a una posición (100,100), desde el tiempo 1 hasta el tiempo 2. Tal ejemplo de elemento de línea de tiempos (el cual puede suministrarse o estar en visualización como parte de una interfaz de usuario) se muestra en la Figura 2, en la cual el objeto "óvalo 1" se establece para que se mueva de una primera posición A (asuma que A corresponde a la posición subyacente (10,10)) hacia una segunda posición B (asuma que B corresponde a la posición subyacente (100,100)). Por lo tanto en el tiempo 1, se genera un marco clave de nivel de objeto para el inicio, y en el tiempo 2, se genera un marco clave de nivel de objeto para el final.

[0029] Ahora asuma que el usuario quiere que el objeto aumente su tamaño en escala a un 150% de su tamaño original, y que regrese a su

tamaño original a lo largo del curso del movimiento desde el tiempo 1 hasta el tiempo 2. De acuerdo con el presente invento, esto puede realizarse al mover el indicador de tiempo actual (como por ejemplo la cabeza reproductora (playhead) 7) al tiempo 1.5, tal y como se muestra en la Figura 3, asegurándose que el elemento de la línea de tiempos esté registrando los cambios, y cambiando el valor de la característica de escala a un 150% en ese punto en el tiempo. En otras palabras, en el tiempo 1.5, se cambia la escala y se crea una instantánea del nivel de objeto. Entonces se utiliza un proceso de ocupación, tal y como el que se describe con respecto a la Figura 4, con el fin de determinar los marcos clave de característica (atributo) en el tiempo actual (tiempo 1.5) en los niveles de objeto adyacentes (aquellos que están justo antes y después, en este caso el tiempo 1 y el tiempo 2). Se pueden utilizar algoritmos de interpolación convencionales.

[0030] Más particularmente, cuando un usuario realiza sus cambios tal y como se indica en la Figura 3, se examina el marco clave compuesto que existe antes de este recién creado marco clave (aquí, el elemento 12 en el tiempo 1), junto con el marco clave compuesto que le sigue inmediatamente al marco clave recién creado en el tiempo (aquí, el elemento 16 en el tiempo 2). Si alguno de éstos marcos clave, o los dos, no tiene(n) marcos clave de atributo para la escala del objeto, entonces se entregan los valores apropiados para la escala para esos marcos clave. En este escenario, se crearán los marcos clave de atributo 22 y 26 respectivamente para un valor de escala del 100% bajo el marco clave compuesto previo, 12 y debajo el próximo marco clave compuesto 16, tal y como se indica en la Figura 3.

[0031] La lógica que le concierne a los ejemplos de los procesos arriba descritos se describe con más detalle con relación al diagrama de flujo de la Figura 4. En la Figura 4, los marcos clave del nivel de objeto cercanos se examinan y se ocupan. Cuando se adiciona un nuevo marco clave, se lleva a cabo un recorrido hacia atrás y hacia delante. Si un marco clave de atributo no existe, y tampoco existe un marco clave de objeto (los dos más temprano y más tarde en el tiempo), entonces se crea un nuevo marco clave de atributo, el cual causa la creación de un marco clave de nivel de objeto.

[0032] Más particularmente, en el paso 405 se cambia un valor de atributo en el modo de animación. En el paso 410 se determina si en el tiempo actual existe o no un nuevo marco clave. De ser así, entonces el valor del marco clave se corrige en el paso 490 y el proceso se completa. Si, sin embargo, en el tiempo actual no existe un marco clave de atributo, en el paso 415 se determina si existe o no un marco clave de objeto posterior al tiempo actual.

51

[0033] Si en el paso 415 no existe un marco clave de objeto más tarde en el tiempo, entonces en el paso 420 se determina si existe o no un marco clave de objeto anterior al tiempo actual. Si no, entonces en el paso 495 se crea un nuevo marco clave de atributo para el tiempo actual, pero si existe un marco clave de objeto antes del tiempo actual, entonces en el paso 425 se determina si hay un marco clave para ese atributo anterior al tiempo actual. Si hay un marco clave de atributo anterior en el tiempo para este atributo, entonces en el paso 455 se mantiene el valor en ese marco clave. Si no hay un marco clave de atributo anterior en el tiempo para este atributo, entonces el valor original de la característica se mantiene en el paso 450 como el valor. En cualquier caso, después del paso 450 o del paso 455, el marco clave de objeto anterior al tiempo esperado se ocupa con el marco clave que contiene el valor correspondiente al valor sostenido o mantenido desde el paso 450 o desde el paso 455 hasta el paso 460. Entonces el procesamiento continúa en el paso 495 con la creación de un nuevo marco clave de atributo para el tiempo actual. Se toma nota que en el caso en que el tiempo actual sea el tiempo = 0, entonces no hay necesidad de mirar más atrás en el tiempo, porque el tiempo = 0 se toma como el menor valor de tiempo posible.

[0034] Si desde el paso 415 se determina que un marco clave de objeto existe en un tiempo posterior al tiempo actual, entonces en el paso 430 se determina si existe o no un marco clave de atributo para este atributo posterior al tiempo actual. Si hay un marco clave de atributo posterior al tiempo actual para este atributo, entonces en el paso 440 el valor en ese marco clave se mantiene en ese marco clave. Si no hay un marco clave de atributo posterior al tiempo actual para este atributo, entonces en el paso 435 el valor utilizado es el valor del último marco clave de atributo registrado, o si no existe ninguno, se utiliza un valor de base (el cual puede ser predeterminado). En cualquier caso, después del paso 435 o del paso 440, el marco clave de objeto más tardío se ocupa con el marco clave que contiene el valor correspondiente al valor sostenido o mantenido desde el paso 435 o desde el paso 440 hasta el paso 445. El procesamiento continúa entonces en el paso 420, tal y como se indicó arriba.

[0035] El método de ocupación de marco clave ejemplar descrito arriba con relación a la Figura 4 ocurre en la creación de un nuevo marco clave. Cuando el indicador del tiempo actual está sobre un marco clave de atributo existente, y ese atributo se modifica mientras se habilita la definición de marcos clave, entonces en ese momento se modifica el valor del marco clave. Esto le permite al sistema mantener las extensiones de tiempo creadas explícitamente por el usuario. Por ejemplo, si el usuario ve los atributos de un objeto de forma explícita y borra el marco clave de atributo que está entre otros dos marcos clave de atributo, entonces esa

extensión de tiempo se mantendrá si se modifica cualquiera de los marcos clave de atributo que limitan la extensión de tiempo.

[0036] Un ejemplo del método para el registro de los valores en los marcos clave le permite al sistema especificar un modo de interacción en donde todas las ediciones que el usuario realiza a cualquier característica generan marcos clave de animación automáticamente. Por ejemplo, un usuario, puede entrar en este modo presionando un botón basculador, el cual bascula al modo de marco clave. Este modo hace posible la creación rápida de un gran número de marcos clave en un periodo de tiempo corto. El número de clics u otros aportes de información requeridos por el usuario se reducen significativamente con relación a otras interfaces de definición de marcos clave.

[0037] Cuando se está en el modo de definición de marcos clave, se vuelve deseable indicarle al usuario que está estableciendo marcos clave con casi cada edición que realiza. Esto se lleva a cabo proveyendo preferiblemente un indicador que esté en contexto con la edición, con el fin de notificarle al usuario que se ha registrado un marco clave para la edición que él acaba de realizar. Este indicador también le da al usuario un botón que puede seleccionar para cancelar el marco clave que se acaba de establecer, o puede escoger otras opciones para interpretar el cambio que se acaba de realizar. Esta interfaz de usuario sobre el objeto se provee en el contexto del cambio, entonces, con respecto a la Figura 5, si el usuario arrastra un objeto 510 en la escena 500 cambiando su posición, la interfaz de usuario (UI) 520 aparecerá en la escena. Por ejemplo, si el usuario cambia el color de un objeto en el color escogiendo la interfaz de usuario (UI), la interfaz de usuario sobre el objeto aparece sobre el editor de color. Una interfaz de usuario (UI) sobre el objeto indica el momento en el que el sistema de animación registra los cambios realizados en el resto de la interfaz de usuario (UI) de la herramienta de autoría. Cuando durante la definición de marcos clave se cambia una característica, se desea que haya un indicador visual para indicar que se ha cambiado un marco clave. Los elementos de información que pueden cambiarse incluyen por ejemplo, x, y, la altura, el ancho, el color, la rotación y el sesgo.

[0038] Existen dos formas distintas de interactuar con objetos en una escena; una de ellas se enfoca en la animación y la otra se enfoca en el diagrama. En el modo "animado", el usuario está particularmente enfocado en la creación de marcos clave para las animaciones. Los cambios que el usuario realiza en las características se registrarán automáticamente como marcos clave en una animación de esas características. En el modo de "diagrama", cuando es posible, los cambios que el usuario realice se interpretarán como cambios a los valores no animados de las características.

[0039] Un ejemplo de una forma de segmentar el grupo de posibles características en un elemento es romperlas en características constituyentes y no constituyentes. Las características constituyentes son aquellas que pueden componerse significativamente a través de operaciones sencillas. Por ejemplo, un flotante es un valor de característica constituyente porque puede componerse por adición con otro valor. Las propiedades no constituyentes no pueden componerse significativamente mediante operaciones simples. Por ejemplo, un valor de color no es un valor constituyente, porque al componerlo a través de una operación tal como la adición se causarían condiciones de gama y otros comportamientos indeseables.

[0040] Una razón para distinguir entre estos dos tipos de características es que, cuando se animan, las características constituyentes pueden cambiarse de forma significativa en el modo de diagrama, mientras que las características no constituyentes no. Por ejemplo, si el usuario tiene una animación en la característica más alta de un elemento (cuyo valor es un flotante) y en ese momento el usuario mueve ese elemento al modo de diagrama, entonces ese cambio se puede componer de forma significativa con la animación (el elemento animará ahora desde su posición actual). Más específicamente, la animación que el usuario especificó en la característica más alta se relaciona al más alto valor actual en el elemento. Esto es posible porque la característica más alta tiene un valor que es constituyente (el motor de la animación toma la posición original del elemento y adiciona el valor de salida de la animación para obtener la posición animada actual del elemento). De este modo, cuando el usuario mueve el elemento al modo de diagrama, el valor de base para la característica más alta se actualiza y el motor de la animación percibirá el cambio automáticamente, y animará desde la nueva posición del elemento.

[0041] Cuando el usuario cambia una característica animada no constituyente en el modo de diagrama, el sistema indica deseablemente que ese cambio no puede incorporarse significativamente con la característica. Por ejemplo, o el usuario se olvidó que está en el modo de diagrama y quiso crear un marco clave en el tiempo actual, o tiene algo más en mente, como cambiar el valor de base de esa característica. En ese caso, el usuario será advertido preferiblemente que el nuevo valor no puede ser incorporado en la animación actual sin crear un marco clave. Adicionalmente, se adiciona deseablemente un indicador a la línea de tiempos para indicar donde ocurrirá la acción, y el valor será deseablemente visible en la escena. Si el usuario entra al modo de diagrama, entonces los valores se convertirán en valores de marco clave. Si el usuario busca la línea de tiempos, entonces los nuevos valores que adicionó se perderán, pues la animación los cancelará.

[0042] Un marco clave compuesto debe representar el estado del objeto al cual pertenece en el momento en que éste se establece en la línea de tiempos. Sin embargo, con el fin de mantener una representación eficiente de las animaciones, solo se almacenan deseablemente los cambios a las características que afectan la animación, y solo éstos se representan en la interfaz del usuario

[0043] En el modo de diagrama, un usuario puede por ejemplo seleccionar adicionar un nuevo marco clave, adicionar una forma, cambiar una característica, o cambiar una selección. Si se debe adicionar un nuevo marco clave, entonces se puede implementar el proceso establecido arriba con respecto a la Figura 4, y se dan más ejemplos abajo (e.g., con respecto a la Figura 9). Si el usuario desea adicionar una forma en un tiempo posterior al 0, entonces el valor base para la opacidad se establece preferiblemente en 0, y un nuevo marco clave se adiciona al tiempo actual, estableciendo la opacidad en 1.

[0044] Si el usuario escoge cambiar una característica, entonces se determina si en el tiempo actual existe o no un marco clave para esa característica. Si es así, entonces el valor del marco clave en el tiempo actual se modifica en consecuencia. Si en el tiempo actual no existe un marco clave para esa característica, entonces se crea un nuevo marco clave de atributo (ver, e.g., la Figura 8).

[0045] Si el usuario escoge hacer un cambio en una selección, se determina si la selección actual tiene o no marcos clave, Si es así, la selección se cambia en consecuencia. Si no, entonces se crea un marco clave compuesto en el tiempo actual.

[0046] De acuerdo con ciertos aspectos de una incorporación, en el modo de diagrama, el usuario puede convertir temporalmente los marcos clave a marcos clave actuales o generar marcos clave temporales. Por ejemplo, si un usuario cambia una característica, entonces, si la característica es constituyente, el valor de base para la característica se modifica deseablemente de tal forma que el valor de la característica sea igual al valor editado en el tiempo actual. Si la característica no es constituyente, y si la característica es animada, entonces se muestran los marcos clave temporales. Sin embargo, si la característica no es constituyente ni animada, entonces el valor de la característica se actualiza deseablemente y el proceso regresa al estado de inicio para el modo de diagrama.

[0047] De forma alterna a los marcos clave temporales, una interfaz de usuario puede proveerle al usuario la opción de editar valores de base (e.g., seleccionando una tabulación de "ninguno" en la interfaz del

5

usuario), o la opción de establecer marcos clave de animación (e.g., seleccionando una tabulación de línea de tiempo). En tal incorporación (cuando el usuario está en la tabulación de "ninguno"), el tiempo actual no está activo (i.e., no está establecido en un valor particular), y el modo de animación no está activo. Se anota que en una tabulación de línea de tiempos específica, tanto el tiempo actual como el modo de animación están activos.

[0048] Un sistema ejemplar comienza en el modo de diagrama. Antes de que las animaciones estén presentes, el usuario seguramente va a arreglar los controles en la forma deseada para el diagrama estático de su interfaz de usuario (UI). Cuando el usuario presiona el botón animado, ingresa deseablemente al modo animado en donde los marcos clave se crean automáticamente para los valores que él cambia. En este modo, se desea que el nuevo botón de marco clave adicionado esté disponible.

[0049] Ahora se describe un ejemplo de creación de animación, con respecto a las Figuras 6A-6F y la Figura 7. Asuma que el usuario desea crear una animación simple que mueve un óvalo de izquierda a derecha a lo largo de medio segundo, mantiene el óvalo detenido por medio segundo y después mueve el óvalo de regreso a su posición original. Inicialmente, el óvalo se crea (paso 700) y se ubica en una posición en la pantalla y se le da un tiempo de inicio (paso 705). Se ingresa el modo animado (paso 710) mediante un clic en un botón animado o realizando una selección de tipo similar. Se crea un nuevo marco clave compuesto para el óvalo en el tiempo actual (paso 715). Por ejemplo en la Figura 6A, el tiempo de inicio para la animación se identifica en la línea de tiempos en el tiempo = 0.5.

[0050] Tal y como se indica en la Figura 6B, la cabeza reproductora (playhead) 600 se mueve hacia delante 0.5 segundos hasta 1 segundo (paso 720). Entonces, como se indica en la Figura 6C, la posición (e.g., posición X,Y) del óvalo se cambia y los marcos clave se adicionan en 1 segundo (paso 725). El procesamiento continúa en el paso 720. La cabeza reproductora (playhead) 600 se mueve hacia delante otros 0.5 segundos, desde 1 segundo hasta 1.5 segundos, tal y como se indica en la Figura 6D, y se adiciona un marco clave (e.g., por un usuario que active sobre la interfaz de usuario (UI) un botón de "adicionar marco clave", por ejemplo), como se muestra en la Figura 6E, para especificar que el óvalo debe mantener su posición entre los segundos 1 y 1.5.

[0051] Entonces, la cabeza reproductora (playhead) 600 se mueve hacia delante otros 0.5 segundos hasta el tiempo = 2, y la posición del óvalo se cambia a su punto de inicio y en consecuencia se adiciona un marco clave, tal y como se muestra en la Figura 6F. En este punto, el usuario ha creado la animación deseada y puede salir del modo animado para cambiar el diagrama de la escena sin crear marcos clave, o puede

quedarse en el modo animado para definir marcos clave de animaciones adicionales.

[0052] Si el usuario está en el modo animado, puede adicionar y ajustar marcos clave en el punto actual en el tiempo al mismo tiempo que ajusta los atributos de objetos, utilizando las asas y las paletas del objeto. Se adiciona un marco clave de atributo para cada atributo que el usuario cambia en un momento dado. Los marcos clave compuestos en el objeto también se tienen en cuenta para conservar su significado. Por otra parte, deseablemente, cuando el usuario cambia una característica y en el tiempo actual hay un marco clave de atributo para esa característica, entonces ese marco clave se edita. Si no, cuando el usuario cambia una característica no hay un marco clave de atributo para esa característica en el tiempo actual, existen varios casos tal y como se establece más adelante, de los cuales se describen ejemplos con respecto a las Figuras 8A-8H.

[0053] Si la cabeza reproductora (playhead) 600 está en el único marco clave compuesto en el elemento, tal y como se indica en la Figura 8A, entonces para el nuevo valor se adiciona un marco clave de atributo en el tiempo actual, tal y como se indica en la Figura 8B.

[0054] Si la cabeza reproductora (playhead) 600 está antes de todos los marcos clave compuestos en el elemento, tal y como se indica en la Figura 8C, entonces en este caso, si el próximo marco clave compuesto no tiene un marco clave compuesto para la característica cambiada, entonces éste se ocupa con el valor viejo de la característica. Tal y como se indica en la Figura 8D, también se crea un marco clave en el tiempo actual con el nuevo valor.

[0055] Si la cabeza reproductora (playhead) 600 se dispone después de todos los marcos clave compuestos en el elemento, tal y como se indica en la Figura 8E, entonces en este caso, si el marco clave compuesto previo no tiene un marco clave de atributo para la propiedad cambiada, entonces se ocupa con el valor viejo de la característica. También se crea un marco clave de atributo en el tiempo actual con un nuevo valor, como se indica en la Figura 8F.

[0056] Si la cabeza reproductora (playhead) 600 se dispone entre dos marcos clave compuestos en el elemento, tal y como se indica en la Figura 8G, entonces en este caso, se verifican tanto el marco clave compuesto anterior al tiempo actual como el marco clave compuesto posterior al tiempo actual. Si ninguno tiene un marco clave de atributo para la característica cambiada, entonces se ocupa con el valor viejo de la característica. También se crea un marco clave de atributo en el tiempo actual con un nuevo valor, como se indica en la Figura 8H. Se anota que el

valor de base no será necesariamente utilizado si hay marcos clave de atributo especificados en cualquier momento más allá de los marcos clave compuestos (e.g., similares a aquellos establecidos arriba con respecto al paso 435 en la Figura 4).

[0057] Un ejemplo de una condición previa a los ejemplos descritos con relación a las Figuras 8A-8H es que se active deseablemente el auto botón de marco clave. Una condición posterior deseable es que los cambios que el usuario le realice a las características se registren como marcos clave de atributos. Los parámetros de los ejemplos descritos con respecto a las Figuras 8A-8H son los que siguen a continuación. Los tiempos y los valores de los atributos se establecen deseablemente mientras se está en el modo de auto definir marcos clave. Si no hubo un marco clave compuesto en el tiempo actual, entonces se crea deseablemente uno y se provee a los parientes de ese elemento para representar el nuevo marco clave de atributo. Si las características no fuesen visibles previamente, entonces se le proveen al usuario par que las vea. Se prevé que este parámetro ejemplar puede ser desactivado como una opción.

[0058] Es deseable que un marco clave compuesto se pueda adicionar a un elemento en el tiempo actual con el fin de preservar el estado mientras se crea una animación. Los marcos clave de espera pueden crearse con esta funcionalidad. Se prevé que un ejemplo de una interfaz de usuario (UI) comprendería un botón de adición de marco clave que se activa cuando se está en el modo de definir/animar marcos clave. Un usuario seleccionaría deseablemente un botón o un selector de "adición de marco clave compuesto" y un nuevo marco clave compuesto se adiciona y se ocupa en el tiempo actual. Si ya existe un marco clave compuesto, entonces no se hará ningún cambio.

[0059] Para ocupar un marco clave compuesto para cada característica que se anima actualmente en el elemento en el cual el marco clave compuesto se está adicionando (tal y como se indica en la Figura 9A), deseablemente, el sistema adiciona una marco clave de atributo que comprende el valor animado actual de la característica. El nuevo marco clave compuesto ejemplar se muestra en la Figura 9B. Deseablemente, el nuevo marco clave compuesto se ocupa con los valores actuales de todas las animaciones en el tiempo actual. Nótese que el usuario no tuvo que cambiar un valor para crear ese marco clave compuesto.

[0060] Una condición previa ejemplar a la creación del marco clave compuesto, es que el usuario busque al tiempo al cual él quiere establecer el marco clave. Por ejemplo, el usuario puede seleccionar "Línea de Tiempos-> Adicione los Marcos Clave" en la interfaz de usuario (UI) o puede presionar el botón de adición de marco de clave en la línea de tiempos del elemento.

[0061] El usuario puede adicionar explícitamente marcos clave de atributo. Esto puede ser deseable si el usuario quisiera un marco clave con el mismo valor que el marco clave actual. En tal caso, en un ejemplo de una interfaz de usuario (UI), el usuario rodaría el tiempo actual al tiempo deseado para el marco clave, movería el ratón a lo largo del área de la línea de tiempos para el tiempo deseado y presionaría el botón de establecimiento del marco clave. De forma alterna, por ejemplo, el usuario puede ser proveído con una opción para seleccionar "adicionar un marco clave" en un menú que se mueva hacia abajo. De este modo se adiciona al atributo un nuevo marco clave de atributo en el tiempo actual.

[0062] De este modo, se contempla que un ejemplo de una interfaz de usuario (UI) puede mantener las preferencias del usuario para los tiempos de marco clave de atributo a lo largo de la ediciones, para atribuirle los valores de marco clave. Por otra parte, los ejemplos de las interfaces de usuario permiten la creación tanto de marcos clave nivel de objeto como de marcos clave de atributo, la ocupación de valores de marco clave de atributo basados en marcos clave de objeto cercanos, y no requieren la creación explícita de los marcos clave para un objeto o atributo particulares.

[0063] Más adelante se contempla que el presente invento puede ser utilizado con transformaciones bidimensionales así como con transformaciones tridimensionales, pero no solo debería limitarse a éstos tipos de transformaciones.

### **Ejemplo de Medio de Computación**

[0064] La Figura 10 ilustra un ejemplo de un medio de sistema de computación 800 adecuado, en el cual el invento puede ser implementado. El medio de sistema de computación 800 es solo un ejemplo de un medio de computación adecuado y su intención no es la de sugerir alguna limitación en lo que concierne el alcance del uso o de la funcionalidad del invento. Tampoco se debe interpretar que el medio de computación 800 tenga alguna dependencia o algún requerimiento relacionado con algún componente o con una combinación de éstos que se ilustre en el ejemplo de un medio operativo 800.

[0065] El invento es operacional con otros numerosos medios o configuraciones de sistema de computación de propósito general o de propósito especial. Ejemplos de sistemas de computación bien conocidos, medios, y/o configuraciones que puedan ser adecuados para el uso con el invento incluyen, pero no se limitan a, computadores personales, computadores de servidor, dispositivos de mano o portátiles, sistemas de multiprocesador, sistemas basados en microprocesadores, cajas de

colocación (*set-top*), aparatos electrónicos de consumidor programables, computadores personales de red, micro computadores, computadores de procesador central, medios de computación distribuidos que incluyen cualquiera de los sistemas o dispositivos anteriores y sus semejantes.

[0066] El invento se puede describir en el contexto general de las instrucciones de computador ejecutables, tales como los módulos de programa que se ejecutan por un computador. Generalmente los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos, etc., que realizan tareas particulares o que implementan tipos de datos abstractos particulares. El invento también puede practicarse en medios de computación distribuidos en donde las tareas se realizan por dispositivos de procesamiento remotos que están conectados a través de una red de comunicaciones u otro medio de transmisión de datos. En un medio de computación distribuido, los módulos de programa y otros datos pueden ser ubicados en medios de almacenamiento de computador, tanto remotos como locales, incluyendo los dispositivos de almacenamiento de memoria.

[0067] Con referencia a la Figura 10, un ejemplo de un sistema para implementar el invento incluye un dispositivo de computación de propósito general en la forma de un computador 810. Los componentes de un computador 810 pueden incluir, pero no se limitan a, una unidad de procesamiento 820, una memoria de sistema 830, y un bus de sistema 821 que asocia varios componentes de sistema, incluyendo la memoria de sistema a la unidad de procesamiento 820. El bus de sistema 821 puede ser cualquiera de los varios tipos de estructuras de bus, incluyendo un bus de memoria o un controlador de memoria, un bus periférico, y un bus local que utilice cualquier arquitectura de bus de una variedad de éstas. Como un ejemplo, y no como una limitación, tales arquitecturas incluyen el bus de la Arquitectura Estándar de Industria (ISA, por sus siglas en inglés), el bus de Arquitectura de Micro Canal (MCA, por sus siglas en inglés) el bus ISA Mejorado (EISA, por sus siglas en inglés), el bus local de la Asociación de los Estándares Electrónicos de Vídeo (VESA, por sus siglas en inglés) y el bus de Interconexión de Componente Periférico (PCI, también conocido como bus de *Mezzanine*).

[0068] El computador 810 incluye típicamente una variedad de medios que puedan ser leídos por computador. Los medios que puedan ser leídos por computador pueden ser cualesquiera medios a los que se pueda entrar por el computador 810, e incluyen tanto los medios volátiles como los no volátiles, y los medios removibles como los no removibles. Como un ejemplo, y no como una limitación, los medios que puedan ser leídos por computador pueden comprender los medios de almacenamiento de computador y los medios de comunicación. Los medios de almacenamiento de computador incluyen tanto los medios volátiles y no volátiles como los

medios removibles y no removibles, implementados en cualquier método o tecnología para el almacenamiento de información, tales como las instrucciones que puedan ser leídas por computador, las estructuras de datos, los módulos de programa y otros datos. El medio de almacenamiento de computador incluye, pero no se limita a, la memoria RAM (memoria de acceso aleatorio), la memoria ROM (memoria sólo de lectura), la memoria EEPROM (memoria sólo de lectura, programable y borrrable en forma eléctrica), la memoria instantánea u otra tecnología de memoria, el CD ROM (memoria de sólo lectura en disco compacto), los discos digitales versátiles (DVD) u otro disco óptico de almacenamiento, los casetes magnéticos, la cinta magnética, el disco de almacenamiento magnético u otros dispositivos de almacenamiento magnético, o cualquier otro medio que pueda ser utilizado para almacenar la información deseada y que se pueda acceder a él por el computador 810. Los medios de comunicación incorporan típicamente instrucciones que puedan ser leídas por computador, estructuras de datos, módulos de programa u otros datos que estén en una señal de datos modulada, tal como una onda portadora u otro mecanismo de transporte e incluye cualesquiera medios de entrega de información. El término "señal de datos modulada" significa una señal que tiene una o más de sus características establecidas o cambiadas de tal forma que se codifica la información en la señal. Como un ejemplo, pero no como una limitación, los medios de comunicación incluyen los medios en cable tales como una red en cable o una conexión en cable directa, y medios inalámbricos tales como los medios acústicos, la RF (radiofrecuencia), los infrarrojos y otros medios inalámbricos. La combinación de cualquiera de los anteriores también debería incluirse dentro del alcance de los medios que puedan ser leídos por computador.

[0069] La memoria de sistema 830 incluye los medios de almacenamiento de computador en la forma de memoria volátil y/o no volátil tal como la memoria ROM 831 y la memoria RAM 832. Un sistema básico de entrada/salida 833 (BIOS), que contiene las rutinas básicas para ayudar en la transferencia de información entre los elementos al interior de un computador 810, tal como durante la puesta en marcha, típicamente se almacena en la memoria ROM 831. Típicamente, la memoria RAM 832 contiene datos y/o módulos de programa que tienen acceso inmediato por una unidad de procesamiento 820 y/o se operan en el presente por ésta. Como un ejemplo, pero no como una limitación, la Figura 10 ilustra la unidad de procesamiento 834, los programas de aplicación 835, otros módulos de programa 836 y los datos de programa 837.

[0070] El computador 810 puede también incluir otros medios de almacenamiento de computador removibles/no removibles, volátiles/no volátiles. Como forma de ejemplo únicamente, la Figura 10 ilustra un impulsor de disco duro 840 que lee desde medios magnéticos no

removibles, no volátiles o escribe en ellos, ilustra un impulsor de disco magnético 851 que lee desde un disco magnético 852 removible, no volátil o escribe en él, y un impulsor de disco óptico 855 que lee desde un disco óptico 856 removible, no volátil o escribe en él, tal como un CD ROM u otros medios ópticos. Otros medios de almacenamiento de computador removibles/no removibles, volátiles/no volátiles que pueden ser utilizados en el medio operativo ejemplar, pero que no se limitan a éstos, son los casetes de cinta magnética, las tarjetas de memoria instantánea, los discos digitales versátiles, las cintas de video digital, la memoria RAM de estado sólido, la memoria ROM de estado sólido y sus semejantes. Típicamente, el impulsor de disco duro 841 se conecta a un bus de sistema 821 a través de una interfaz de memoria no removible tal como la interfaz 840, y el impulsor de disco magnético 851 y el impulsor de disco óptico 855 típicamente se conectan al bus de sistema 821 por una interfaz removible, tal como la interfaz 850.

[0071] Los impulsores y sus medios de almacenamiento de computador asociados proveen el almacenamiento de las instrucciones que puedan ser leídas por computador, las estructuras de datos, los módulos de programa y otros datos para el computador 810. Por ejemplo, en la Figura 10, el impulsor de disco duro 841 se ilustra como un sistema operativo de almacenamiento 844, como programas de aplicación 845, como otros módulos de programa 846 y como datos de programa 847. Nótese que estos componentes pueden ser los mismos que un sistema operativo de almacenamiento 834, que los programas de aplicación 835, que otros módulos de programa 836 y que los datos de programa 837, o ser diferentes de estos. El sistema operativo de almacenamiento 844, los programas de aplicación 845, los otros módulos de programa 846 y los datos de programa 847 aquí tienen números diferentes para ilustrar que, como mínimo, son copias diferentes. Un usuario puede ingresar los comandos y la información en el computador 810 a través de dispositivos de entrada tales como un teclado 862 y un dispositivo indicador 861, comúnmente mencionado como un "ratón", una bola de control, o una tablilla de graficación tacto-sensible. Otros dispositivos de entrada (no mostrados) pueden incluir un micrófono, una palanca de mando, un teclado de juego, un plato satélite, un escáner o sus semejantes. Estos y otros dispositivos de entrada se conectan generalmente a la unidad de procesamiento 820 a través de una interfaz de entrada de usuario 860 que se asocia a un bus de sistema, pero que puede conectarse por otras estructuras de interfaz o de bus, tales como un puerto paralelo, un puerto de juego o un bus serial universal (USB). También se conecta un monitor 891 u otro tipo de dispositivo de visualización al bus de sistema 821 por medio de una interfaz, tal como una interfaz de video 890. En adición al monitor, los computadores también pueden incluir otros dispositivos periféricos de salida tales como los parlantes 897 y la impresora 896, los

62

cuales se pueden conectar a través de una interfaz periférica de salida 895.

[0072] El computador 810 puede operar en un medio que trabaje en red utilizando conexiones lógicas a uno o más computadores remotos, tales como un computador remoto 880. El computador remoto 880 puede ser un computador personal, un servidor, un selector de vía, un PC en red, un dispositivo igual u otro nodo en red común, y típicamente incluye muchos de los elementos arriba descritos relacionados con el computador 810 o todos ellos, aunque en la Figura 10 solo se ilustró un dispositivo de almacenamiento de memoria 881. Las conexiones lógicas descritas incluyen una red LAN (red de área local) 871 y una red WAN (red de área de trabajo) 873, pero también pueden incluir otras redes. Tales medios que trabajan en red son comunes en las oficinas, en redes de grandes computadores de empresa, en intranets y la Internet.

[0073] Cuando el computador 810 se utiliza en un medio que trabaja con la red LAN, éste se conecta a la red LAN 871 a través de una interfaz de red o de un adaptador 870. Cuando el computador 810 se utiliza en un medio que trabaja con la red WAN, éste típicamente incluye un módem 872 u otros medios para establecer las comunicaciones sobre la red WAN 873, tal como la Internet. El módem 872, el cual puede ser externo o interno, puede conectarse al bus de sistema 821 por medio de una interfaz de entrada de usuario 860, u otro mecanismo apropiado. En un medio que trabaja en red, los módulos de programa descritos con relación al computador 810, o partes de éstos, se pueden almacenar en el dispositivo de almacenamiento de memoria remoto. Como un ejemplo, pero no como una limitación, la Figura 10 ilustra los programas de aplicación remotos 885 residiendo en el dispositivo de memoria 881. Se apreciará que las conexiones de red mostradas son un ejemplo y que se pueden utilizar otros medios para establecer un enlace de comunicación entre los computadores.

[0074] Como se mencionó arriba, mientras que las incorporaciones de ejemplo del presente invento se han descrito en conexión con varios dispositivos de computación, los conceptos subyacentes pueden aplicarse a cualquier dispositivo o sistema de computación.

[0075] Las diferentes técnicas aquí descritas pueden implementarse en conexión con el soporte físico (hardware) o con el software o, cuando es adecuado, con una combinación de los dos. De este modo, los métodos y los aparatos del presente invento, o ciertos aspectos o porciones de éste, pueden tomar la forma de un código de programa (i.e., instrucciones) incorporado en medios tangibles, tales como disquetes flexibles, CD ROMs, discos duros, o cualquier otro medio de almacenamiento que pueda ser leído por una máquina, en los que, cuando el código de programa se carga

63

en una máquina, tal como un computador, y es ejecutado por ésta, la máquina se convierte en un aparato para la práctica del invento. En el caso de la ejecución del código de programa en computadores que pueden programarse, el dispositivo de computación incluirá generalmente un procesador, un medio de almacenamiento que puede ser leído por el procesador (incluyendo los elementos de almacenamiento y/o memoria volátiles y no volátiles), por lo menos un dispositivo de entrada, y por lo menos un dispositivo de salida. Si se desea, el o los programa(s) puede(n) implementarse en lenguaje ensamblador o de máquina. En cualquier caso, el lenguaje puede ser un lenguaje compilado o interpretado, y puede combinarse con las implementaciones del soporte físico (hardware).

[0076] Los métodos o los aparatos del presente invento también se pueden practicar por medio de las incorporaciones de las comunicaciones en la forma de un código de programa que se transmite por algún medio de transmisión, tal como por el cableado o el conexionado eléctrico, a través de fibra óptica, o por medio de cualquier otra forma de transmisión, en la que, cuando el código de programa se recibe, se carga y se ejecuta en una máquina, tal como una EPROM (memoria sólo de lectura, programable y borrrable), como una red de compuertas, como un dispositivo lógico programable (PLD), como un computador de un cliente o sus semejantes, la máquina se convierte en un aparato para la práctica del invento. Cuando el código de programa se implementa en un procesador para propósitos generales, éste código de programa se combina con el procesador, para proveer un aparato único que opera para invocar la funcionalidad del presente invento. Adicionalmente cualesquiera técnicas de almacenamiento utilizadas en conexión con el presente invento, pueden ser invariablemente una combinación del soporte físico (hardware) y del software.

[0077] Cuando el presente invento se describió en conexión con las incorporaciones preferidas de las diferentes figuras, debe entenderse que se pueden utilizar otras incorporaciones similares, o que se pueden realizar modificaciones o adiciones a las incorporaciones descritas para llevar a cabo la misma función del presente invento, sin desviarse de él. Por lo tanto, el presente invento no se debería limitar a una única característica, y más bien se debería interpretar en el alcance y en la amplitud de acuerdo con las reivindicaciones anexas.

## LO QUE SE REIVINDICA ES:

1. Un método para definir marcos clave para un objeto, que comprende:  
la identificación de, al menos, una característica y de un tiempo para el objeto;  
la creación de un primer marco clave compuesto en el tiempo;  
la recepción de un segundo tiempo para el objeto; y  
la creación de un segundo marco clave compuesto en el segundo tiempo.
2. El método de la reivindicación 1, que comprende además la recepción de tiempos adicionales para el objeto y la creación de marcos clave compuestos asociados en cada uno de los tiempos adicionales.
3. El método de la reivindicación 1, en el que la recepción del segundo tiempo para el objeto comprende el desplazamiento de una cabeza reproductora (playhead) a una posición en la línea de tiempos en la interfaz de usuario, la posición correspondiente al segundo tiempo.
4. El método de la reivindicación 1, que comprende además el ingreso a un modo animado antes de la creación del primer marco clave compuesto.
5. El método de la reivindicación 1, en el que cada uno de los marcos clave compuestos, ya sea el primero o el segundo, representan el estado de al menos una característica en el objeto en el tiempo asociado.
6. El método de la reivindicación 1, que más adelante comprende la recepción de un cambio de al menos una característica, antes de la creación del segundo marco clave compuesto, el segundo marco clave compuesto que incorpora el cambio a por lo menos una característica.
7. El método de la reivindicación 6, que más adelante comprende la creación de un marco clave de atributo sensible al cambio recibido por al menos una característica si no existe un marco clave de atributo para al menos una característica en el tiempo en el que se recibe el cambio recibido, y el cambio de un marco clave de atributo existente sensible al cambio recibido al menos por una característica si el marco clave de atributo existente existe en el tiempo en el que se recibe el cambio recibido.
8. Un método de definir marcos clave para un objeto, que comprende:  
la recepción de un valor para un atributo para un objeto en el primer tiempo;  
si en el primer momento existe un marco clave de atributo correspondiente al atributo, entonces se corrige el marco clave de atributo sensible al valor recibido para el atributo;

aparte de eso, si un marco clave de objeto existe, se ocupa un marco clave del objeto cercano con un marco clave de atributo.

9. El método de la reivindicación 8, que comprende además la creación de un marco clave de atributo en el primer tiempo si no existe un marco clave de atributo ni tampoco existe un marco clave de objeto en el primer momento.

10. El método de la reivindicación 8, en el que la ocupación de un marco clave del objeto cercano comprende:

si el marco clave del objeto cercano existe en el tiempo después del primer tiempo, y un marco clave de atributo existe en el tiempo después del primer tiempo, entonces se establece un primer valor al valor del marco clave de atributo que existe en el tiempo después al primer tiempo;

si el marco clave del objeto cercano existe en el tiempo después del primer tiempo, y un marco clave de atributo no existe en el tiempo después del primer tiempo, entonces se establece el primer valor al valor del último marco clave de atributo registrado, o a un valor de base si no existe ninguno;

la creación de un nuevo marco clave de atributo en el marco clave del objeto cercano utilizando el primer valor.

11. El método de la reivindicación 8, que más adelante comprende:

si la clave de objeto que existe es anterior en el tiempo al primer momento, y un marco clave de atributo existe en el tiempo antes del primer momento, entonces se establece un segundo valor al valor del marco clave de atributo que existe en el tiempo antes del primer momento;

si la clave de objeto que existe es anterior en el tiempo al primer momento, y un a marco clave de atributo no existe en el tiempo antes del primer momento, entonces se establece el segundo valor al valor correspondiente al valor recibido originalmente para el atributo; y

la creación de un nuevo marco clave de atributo en el marco clave del objeto cercano, utilizando el segundo valor.

12. El método de la reivindicación 8, en el que la ocupación de un marco clave del objeto cercano comprende:

si la clave de objeto que existe es anterior en el tiempo al primer momento, y un marco clave de atributo existe en el tiempo antes del primer momento, entonces se establece un primer valor al valor del marco clave de atributo que existe en el tiempo antes del primer momento;

si la clave de objeto que existe es anterior en el tiempo al primer momento, y un a marco clave de atributo no existe en el tiempo antes del primer momento, entonces se establece el primer valor al valor correspondiente al valor recibido originalmente para el atributo; y

la creación de un nuevo marco clave de atributo en el marco clave del objeto cercano utilizando el primer valor.

13. En un sistema de computador que tiene una interfaz gráfica de usuario que incluye un visualizador y un dispositivo de selección de interfaz de usuario, un método de definir marcos clave para un objeto por medio de un elemento de línea de tiempos en el visualizador, que comprende:

la recepción de una señal de selección que indica que el dispositivo de selección de interfaz de usuario está seleccionando al menos una característica y un tiempo para el objeto;

el visualizador de un primer marco clave compuesto en el tiempo en el elemento de línea de tiempos;

la recepción de una señal de selección que indica que el dispositivo de selección de interfaz de usuario está seleccionando un segundo tiempo para el objeto; y

la visualización de un segundo marco clave compuesto en el segundo tiempo en el elemento de línea de tiempos.

14. El método de la reivindicación 13, que comprende además la recepción de señales de selección adicionales que indican que el dispositivo de selección de interfaz de usuario está seleccionando tiempos adicionales para el objeto, y el visualizador de los marcos clave compuestos asociados en cada uno de los tiempos adicionales en el elemento de la línea de tiempos.

15. El método de la reivindicación 13, en el que recibir la señal de selección que indica que el dispositivo de selección de interfaz de usuario está seleccionando un segundo tiempo para el objeto, comprende la recepción de una señal de ejecución, que indica que un usuario está moviendo una cabeza reproductora (playhead) a una posición en la línea de tiempos en el elemento de línea de tiempos, la posición que corresponde al segundo tiempo.

16. El método de la reivindicación 13, que comprende además la recepción de una señal ejecutiva, que indica que un usuario está seleccionando un modo animado antes del visualizador del primer marco clave.

17. El método de la reivindicación 13, que comprende además la recepción de una señal de selección que indica que el dispositivo de selección de interfaz de usuario está seleccionando un cambio en al menos una característica antes del visualizador del segundo marco clave compuesto, el segundo marco clave compuesto que incorpora el cambio de al menos una característica.

18. El método de la reivindicación 17, que comprende además el visualizador de un marco clave de atributo sensible al cambio recibido al menos por una característica en el elemento de la línea de tiempos si no existe un marco clave de atributo para al menos una característica en el tiempo en que se recibe el cambio recibido, y el cambio de un marco clave de atributo nuevo que está en el visualizador y que es sensible al cambio recibido al menos por una característica, si el marco clave de atributo existente que está en el visualizador existe en el momento en que se recibe el cambio recibido.

19. Un dispositivo de visualización que reproduce un elemento de línea de tiempos para definir marcos clave para un objeto mediante la recepción de una señal de selección que indica la selección de al menos una característica y de un tiempo para el objeto; el visualizador de un primer marco clave compuesto en el tiempo en el elemento de línea de tiempos; la recepción de una señal de selección que indica la selección de un segundo tiempo para el objeto; el visualizador de un segundo marco clave compuesto que está en visualización en el segundo tiempo en el elemento de línea de tiempos.

20. El dispositivo de visualización de la reivindicación 19, en el que el elemento de la línea de tiempos se adapta para recibir señales de selección adicionales que indican la selección de tiempos adicionales para el objeto, y el visualizador asociado a los marcos clave compuestos en cada uno de los tiempos adicionales en el elemento de línea de tiempos.

21. El dispositivo de visualización de la reivindicación 19, en el que el elemento de la línea de tiempos comprende una cabeza reproductora (playhead) que se puede mover.

22. El dispositivo de visualización de la reivindicación 19, en el que el elemento de la línea de tiempos se adapta para recibir una señal de ejecución que indica que un usuario está seleccionando un modo animado.

23. El dispositivo de visualización de la reivindicación 19, en el que el elemento de la línea de tiempos se adapta para recibir una señal de selección que indica la selección de un cambio de al menos una característica antes de hacer la visualización del segundo marco clave compuesto, el segundo marco compuesto que incorpora el cambio al menos de una característica.

24. El dispositivo de visualización de la reivindicación 23, en el que además el elemento de línea de tiempos se adapta para hacer la visualización de un marco clave de atributo sensible al cambio recibido por al menos una característica en el elemento de línea de tiempos si no existe

un marco clave de atributo para al menos una característica en el tiempo en que se recibe el cambio recibido, y para cambiar un marco clave de atributo existente que está en visualización y que es sensible al cambio recibido al menos por una característica si el marco clave de atributo existente que está en visualización existe en el tiempo en que se recibe el cambio recibido.

25. Un método en un sistema de computador para visualizar en un dispositivo de visualización la definición de marcos clave de un objeto, el cual comprende:

la recepción de una indicación producida por un usuario de un valor para un atributo para el objeto en un primer tiempo.

si en el primer tiempo existe un marco clave de atributo que corresponde al atributo, de acuerdo con la indicación de usuario recibida, entonces en el dispositivo de visualización se realiza la visualización del marco clave de atributo sensible al valor recibido para el atributo;

por lo demás, si existe un marco clave de objeto, en el dispositivo de visualización se realiza la visualización de un marco clave de atributo cercano con un marco clave de atributo.

26. El método de la reivindicación 25, que comprende además un marco clave de atributo en el primer tiempo si en el primer momento no existen ni un marco clave de atributo ni tampoco un marco clave de objeto.

27. El método de la reivindicación 25, en el que hacer la visualización del marco clave del objeto cercano en el dispositivo de visualización comprende:

si el marco clave del objeto cercano existe en el tiempo después del primer momento, y un marco clave de atributo existe en el tiempo después del primer momento, entonces se establece un primer valor al valor del último marco clave de atributo que existe en el tiempo después del primer momento;

si el marco clave del objeto cercano existe en el tiempo después al primer momento, y un marco clave de atributo no existe en el tiempo después del primer momento, entonces se establece el primer valor al valor del último marco clave de atributo registrado o a un valor de base, si no existe ninguno; y

realizar la visualización de un nuevo marco clave de atributo en el marco clave del objeto cercano utilizando el primer valor.

28. El método de la reivindicación 27, que comprende además:

si la clave de objeto que existe es anterior en el tiempo al primer momento, y un marco clave de atributo existe en el tiempo antes del primer momento, entonces se establece un segundo valor al valor del marco clave de atributo que existe en el tiempo antes del primer momento;

si la clave de objeto que existe es anterior en el tiempo al primer momento, y un marco clave de atributo no existe en el tiempo antes del primer momento, entonces se establece el segundo valor al valor correspondiente al valor recibido originalmente para el atributo; y

realizar la visualización de un nuevo marco clave de atributo en el marco clave del objeto cercano utilizando el segundo valor.

29. El método de la reivindicación 25, en el que hacer la visualización del marco clave del objeto cercano en el dispositivo de visualización comprende:

si la clave de objeto que existe es anterior en el tiempo al primer momento, y un marco clave de atributo existe en el tiempo antes del primer momento, entonces establecer un primer valor al valor del marco clave de atributo que existe en el tiempo antes del primer momento;

si la clave de objeto que existe es anterior en el tiempo al primer momento, y un a marco clave de atributo no existe en el tiempo antes del primer momento, entonces establecer el primer valor al valor correspondiente al valor recibido originalmente para el atributo; y

realizar la visualización de un nuevo marco clave de atributo en el marco clave del objeto cercano utilizando el primer valor.

30. En un sistema de computador que tiene una interfaz gráfica de usuario que incluye una visualización y un dispositivo de selección de interfaz de usuario, un método de proveer una interfaz de usuario sobre el objeto en la visualización, que comprende:

la recepción de una señal de selección que indica que el dispositivo de selección de interfaz de usuario está seleccionando un objeto;

la recepción de una señal de selección que indica que el dispositivo de selección de interfaz de usuario está seleccionando una característica del objeto para cambiar;

realizar la visualización de una interfaz de usuario en la proximidad del objeto.

31. El método de la reivindicación 30, en el que la interfaz de usuario que está en visualización indica el momento en que un sistema de animación registra los cambios realizados en una interfaz de usuario de herramienta de autoría.

## **RESUMEN DE LA REVELACIÓN**

Se proveen los sistemas, métodos e interfaces de usuario para la definición de marcos clave de nivel de característica y de nivel de objeto combinado, en los cuales un usuario puede ingresar al modo en donde se registra como una acción de definir marcos clave (e.g., creación o edición) cualquier cambio en una característica en el tiempo actual tal como lo indica una línea de tiempos. Cuando se crea un marco clave en este modo, los marcos clave de nivel de objeto vecinos se inspeccionan para garantizar que se mantiene la funcionalidad que el usuario espera de los marcos clave de nivel de objeto. Si un usuario crea explícitamente una animación en un atributo que se extiende sobre un marco clave de nivel de objeto, entonces esa expansión se mantiene para la animación a medida que el usuario edita.

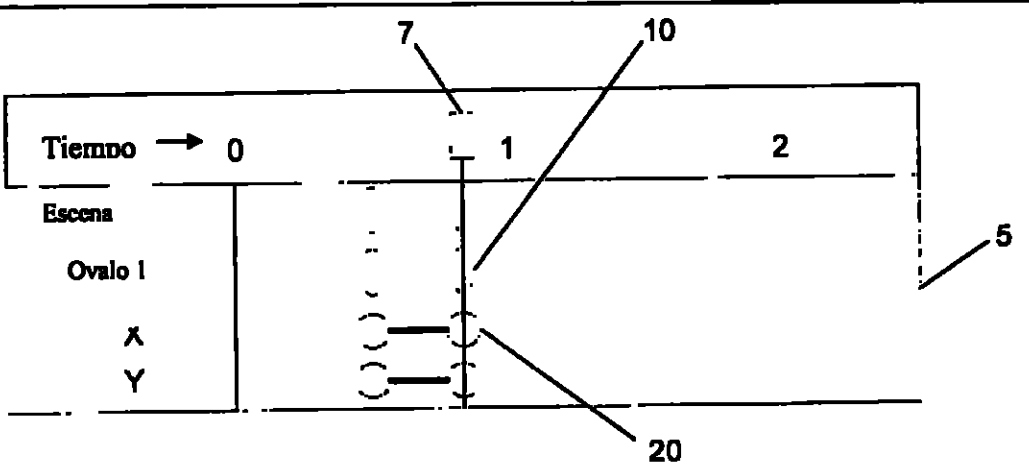


Fig. 1

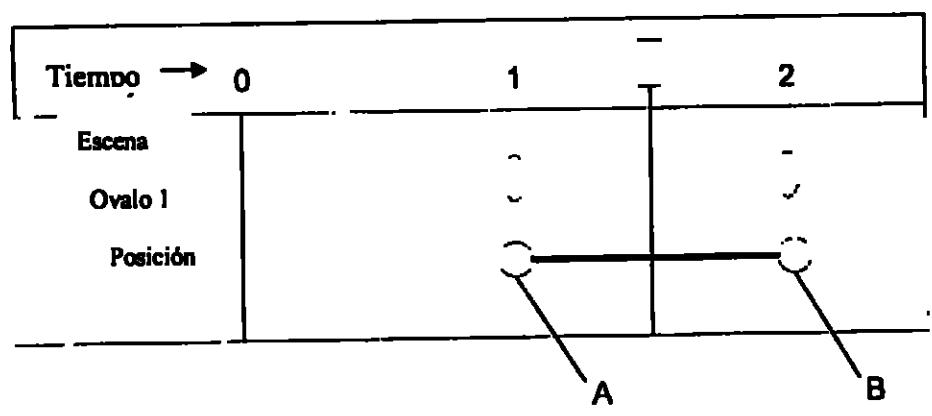


Fig. 2

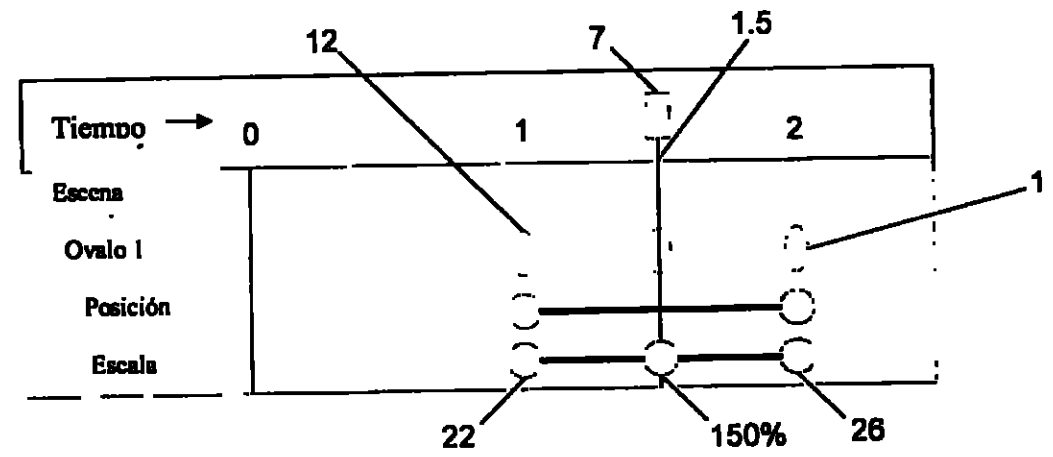


Fig. 3

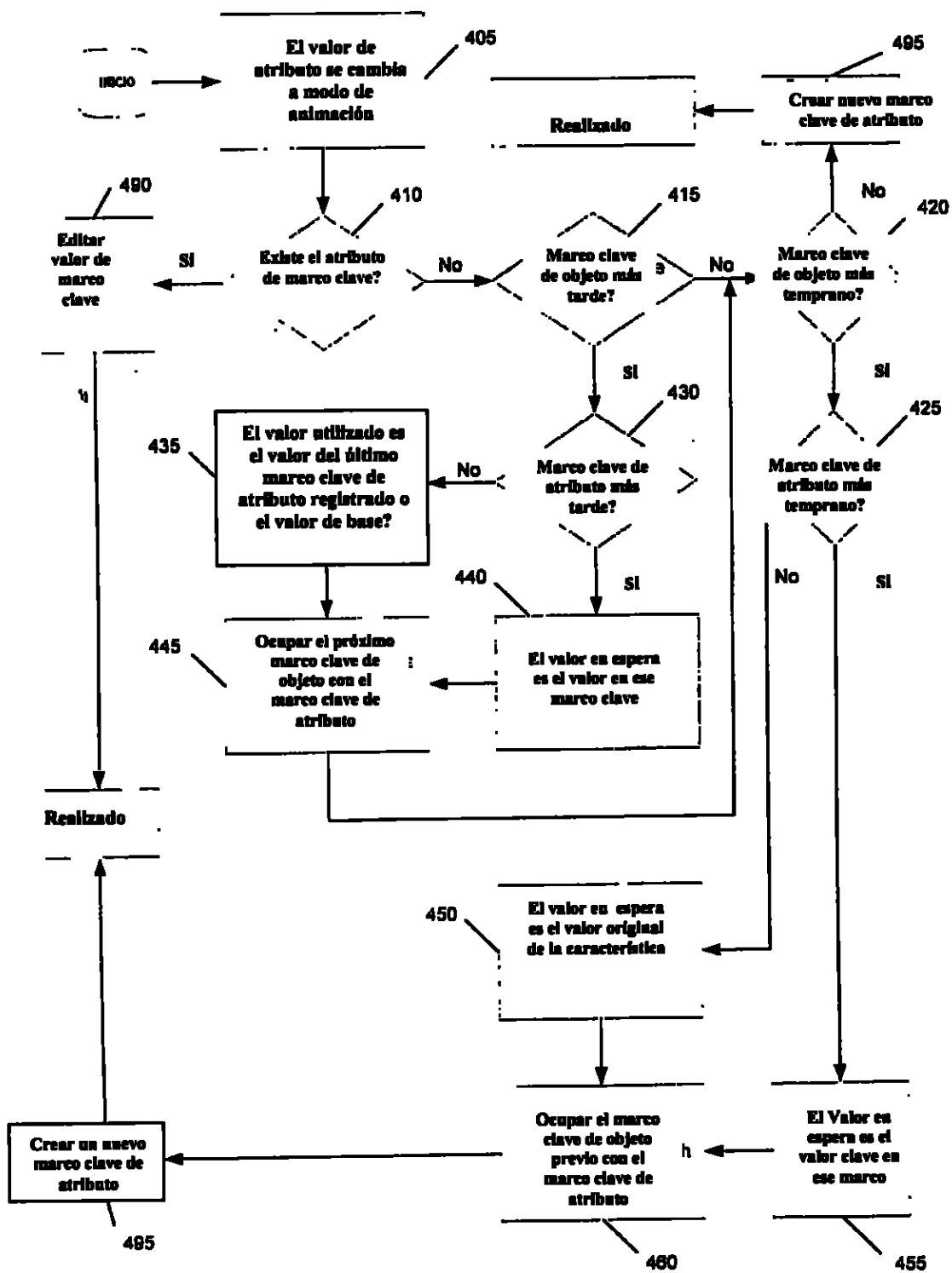
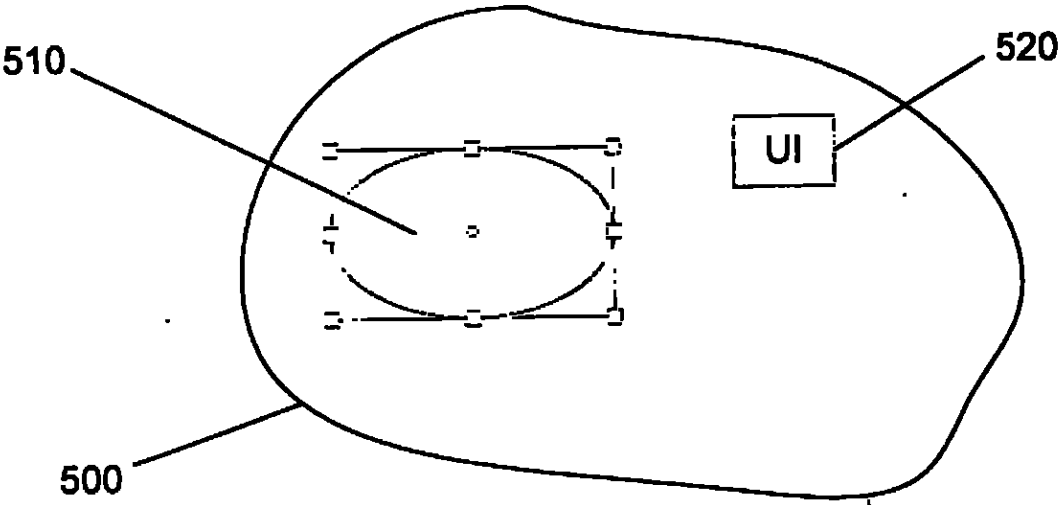


Fig. 4



**Fig. 5**

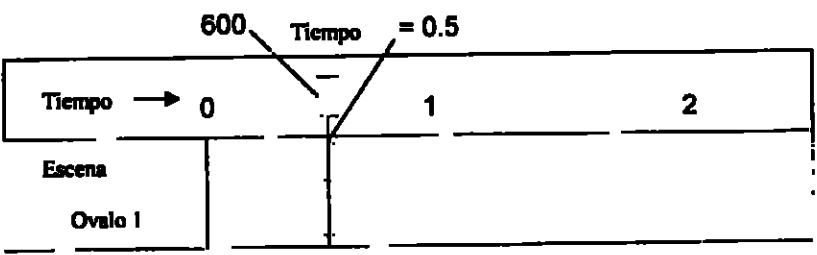


Fig. 6A



Fig. 6B

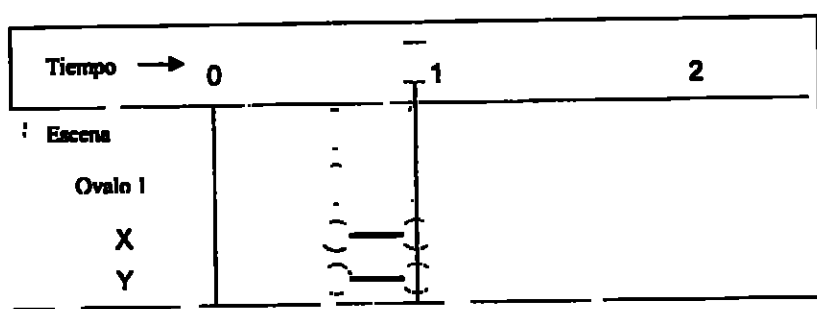


Fig. 6C

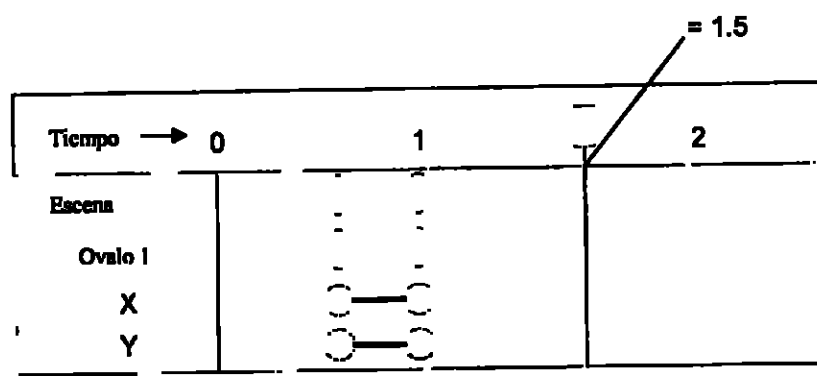


Fig. 6D

75

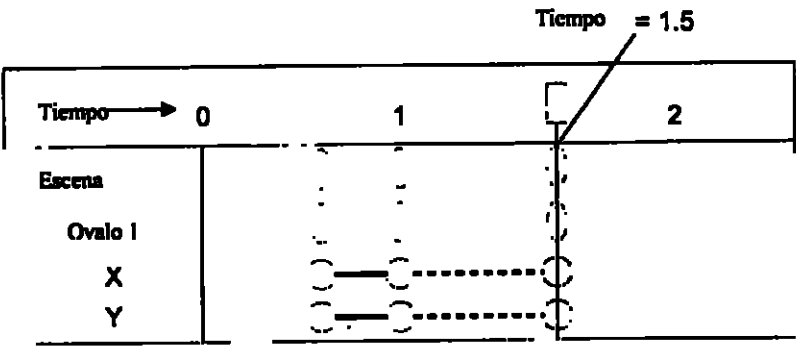


Fig. 6E

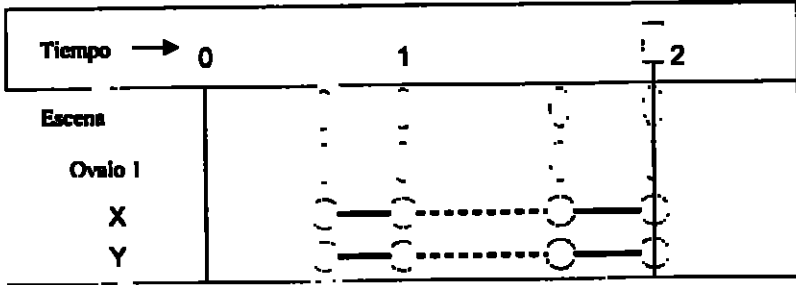


Fig. 6F

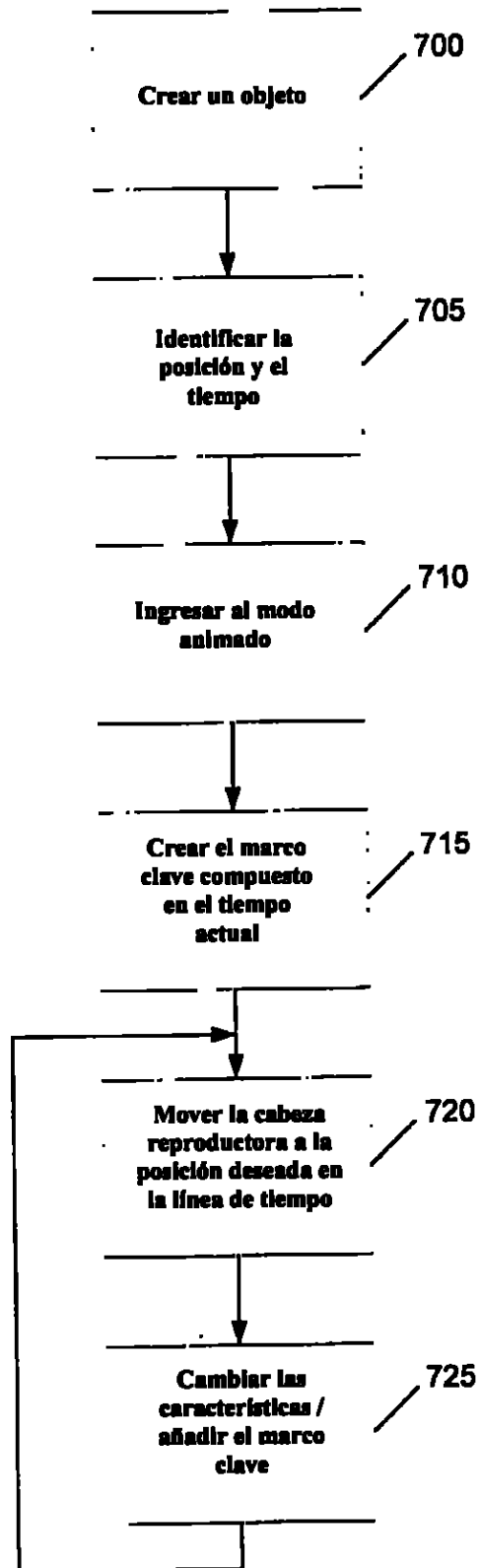


Fig. 7

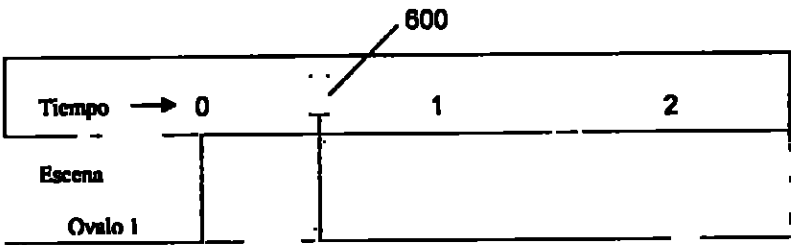


Fig. 8A

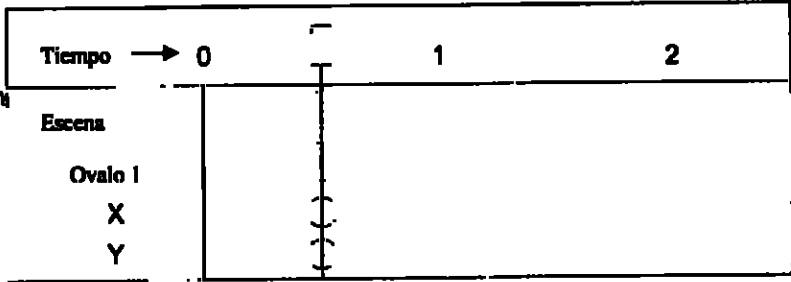


Fig. 8B

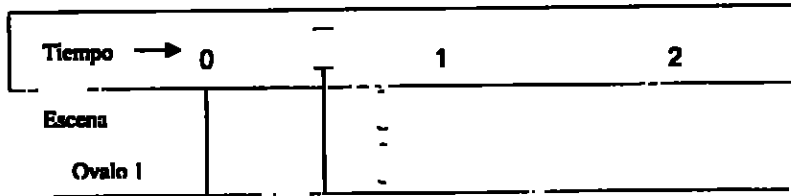


Fig. 8C

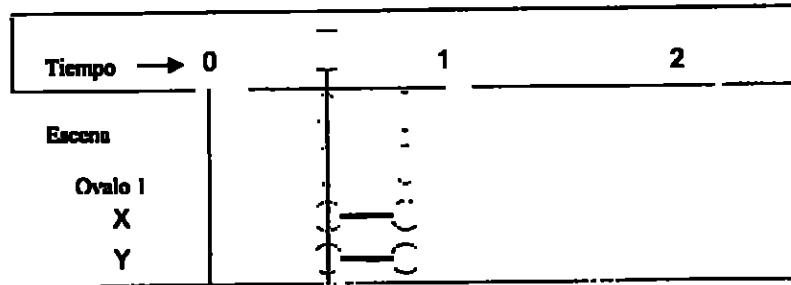


Fig. 8D

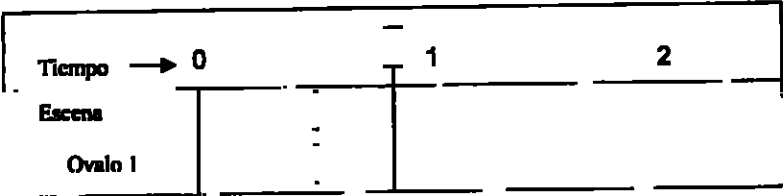


Fig. 8E

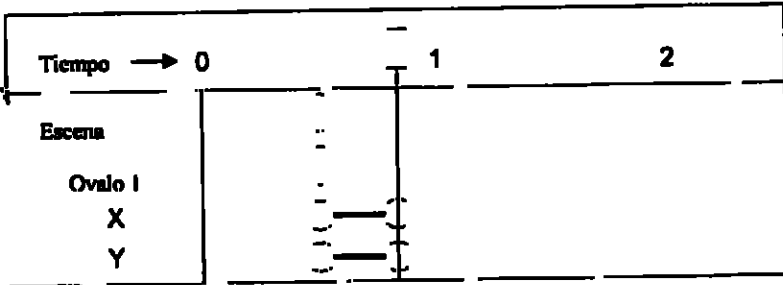


Fig. 8F

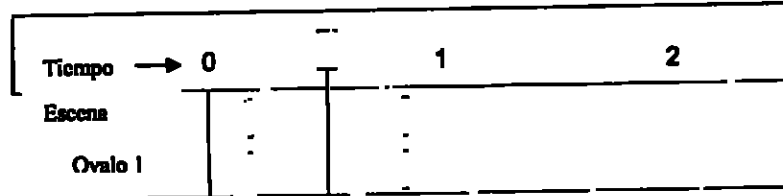


Fig. 8G

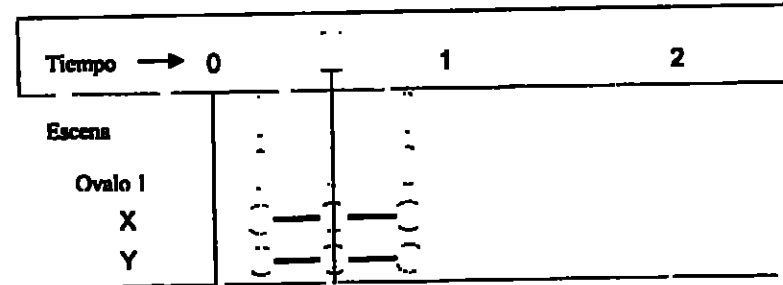


Fig. 8H

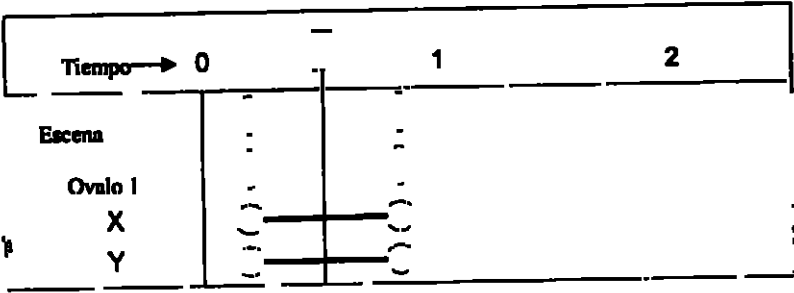


Fig. 9A

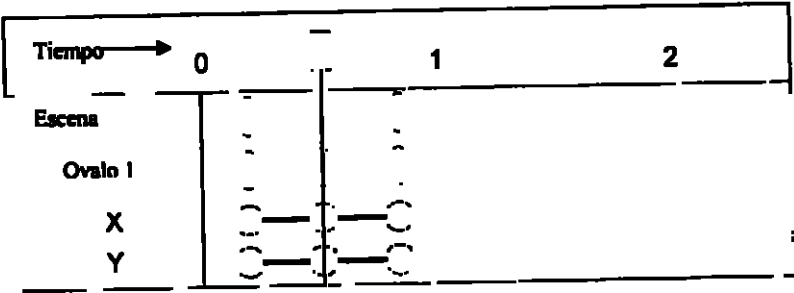


Fig. 9B

Entorno de Computación  
800

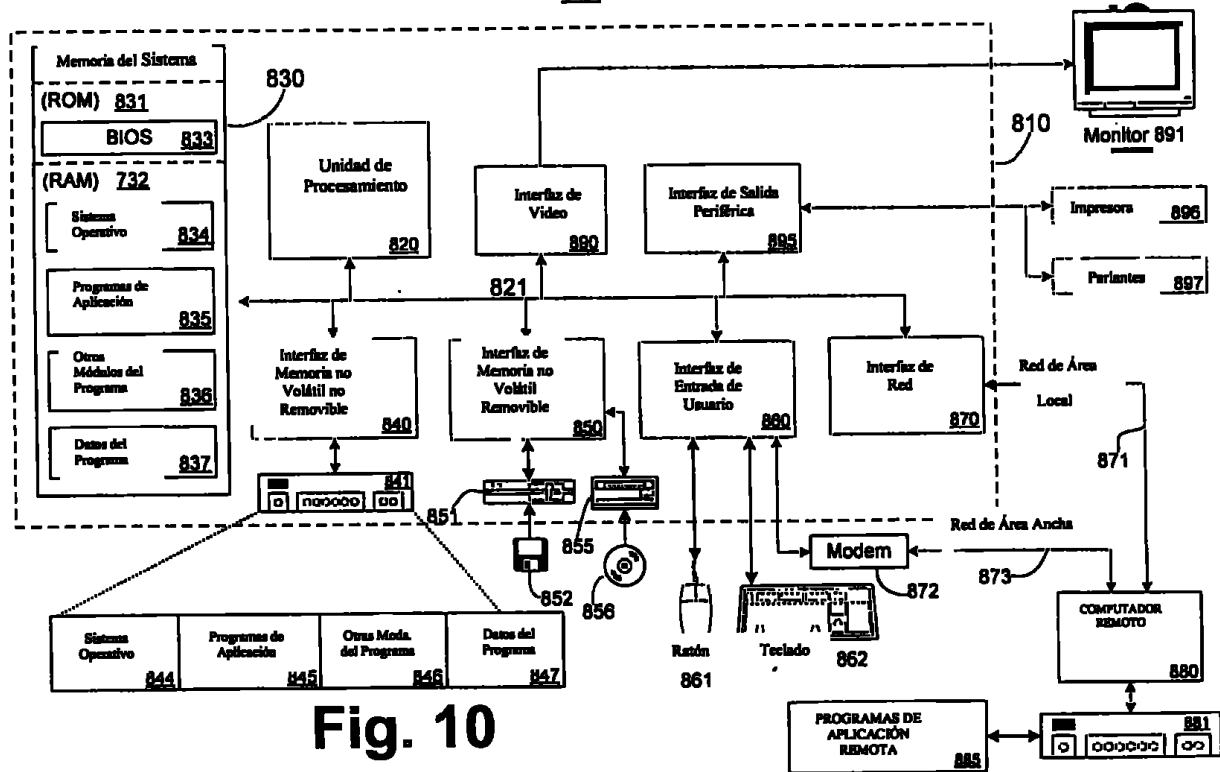


Fig. 10

9

82  
81

## REQUISITOS NECESARIOS PARA PRESENTACION Y AGILIZACION DEL TRAMITE

### PATENTE DE INVENCION

### MODELO DE UTILIDAD

USUARIO      RADICADOR

- Indicación de que se solicita la concesión de una patente.
- Datos de identificación del solicitante o de la persona que se presenta la solicitud.
- Descripción de la invención.
- Dibujos de ser estos pertinentes.
- Comprobante de Pago de las tasas establecidas.

( )      ( )  
( )      ( )  
( )      ( )  
( )      ( )

COMPLETA      ( )

INCOMPLETA      ( )      ( )

### DISEÑO INDUSTRIAL      ( )

- Indicación de que se solicita el registro de Diseño Industrial
- Datos de identificación del solicitante o de la persona que presenta la solicitud.
- Representación gráfica o fotográfica del Diseño Industrial o muestra del Material que incorpora el diseño.
- Comprobante de pago de las tasas establecidas

( )      ( )  
( )      ( )  
( )      ( )  
( )      ( )

COMPLETA      ( )

INCOMPLETA      ( )      ( )

### ESQUEMA DE TRAZADO      ( )

- Indicación de que solicita el registro de un Esquema de trazado.
- Datos de identificación del solicitante o de la persona que presenta La solicitud.
- Representación gráfica del esquema de trazado
- Comprobante de pago de las tasas establecidas.

( )      ( )  
( )      ( )  
( )      ( )  
( )      ( )

COMPLETA

INCOMPLETA      ( )      ( )

Firma Responsable:



Fecha:

SUPERINTENDENCIA Y COMERCIO  
Radicación : 05021155 00000000      Folios: 01  
Fecha (HUB): 2005-05-08 12:27:22  
Trámite : 002 PRELIMINAR Y REGISTRO/ 411 PRESENTAC  
Dependencia: 2009 DIVISION DE NUEVOS MARCAS



62

# Industria y Comercio

## SUPERINTENDENCIA

Folio 88

2020  
Bogotá, D. C.

Doctor(a)  
DARIO CARDENAS NAVAS  
Apoderado(a)

Oficio No.  
12541

|             |            |          |
|-------------|------------|----------|
| REFERENCIA: | Radicación | 05-21195 |
|             | Trámite    | 002      |
|             | Evento     | 001      |
|             | Actuación  | 430      |
|             | Folios     | 001      |

Como resultado del examen de forma, realizado con fundamento en el artículo 38 de la Decisión 486 de la Comisión de la Comunidad Andina, se le comunica que:

- Allegar copia del documento en que conste la cesión del derecho a la patente otorgado por el inventor al solicitante o a su causante. (art. 26-k, Dec 486/2000).

En cumplimiento del artículo 39 de la Decisión 486, se le requiere para que dentro del término de dos (2) meses siguientes a la fecha de notificación del presente oficio, complete los requisitos anteriormente enunciados. En caso de no dar cumplimiento al presente requerimiento, la solicitud se considerará abandonada y perderá su prelación.

Notifíquese el presente oficio conforme a lo dispuesto en el numeral 5.2, literal e), del capítulo quinto del título primero de la Circular Unica No.10 de 2001.

**ALIX CARMENZA CÉSPEDES DE VERGEL**  
Jefe de la División de Nuevas Creaciones

14 ABR 2005

El anterior requerimiento fue notificado por fijación en lista de fecha \_\_\_\_\_

Se recuerda al interesado que debe presentar, dentro de los dieciséis meses siguientes contados a partir de la fecha de presentación de la solicitud cuya prioridad se invoca, copia de la misma certificada por la autoridad que la expidió y, en los casos en que haya lugar, la traducción simple del capítulo reivindicatorio. El incumplimiento de este plazo, acarreará la pérdida de la prioridad invocada.

202027

Carrera 13 No. 27-00 Pisos 5, 7 y 10  
Conmutador: 382 0840  
Fax: 350 5220  
E-mail: Info slc.gov.co  
www.slc.gov.co  
Bogotá, D.C., Colombia