

Kiosco Informativo Interactivo

Informational Kiosk Interactive

Nadia Lee García *, Gisela T. de Clunie #, Gregorio Santana ^

*Universidad Tecnológica de
Panamá
CIDITIC
Panamá, Rep. de Panamá
nadia.lee@utp.ac.pa

#Universidad Tecnológica de
Panamá
Facultad de Ingeniería de Sistemas
Panamá, Rep. de Panamá
gisela.clunie@utp.ac.pa

^Universidad Tecnológica de
Panamá
CIDITIC
Panamá, Rep. de Panamá
gregorio.santana@utp.ac.pa

Abstract: This paper presents the implementation of the System user interface KINFO-UTP using video game technology. We discuss the cataloging facilities building # 3 of the Technological University of Panama, which are displayed in the system. We report the use case diagram KINFO-UTP system and discuss how to combine four programming languages to enable interactivity KINFO-UTP.

Keywords: human-computer interaction, user experience, new interaction paradigms, gaming, 3D game engines, 3D modeling, animation, programming.

Resumen: Este artículo presenta la implantación de la interface de usuario del Sistema KINFO-UTP mediante tecnología de videojuegos. Se discute la catalogación de las instalaciones del edificio #3 de la Universidad Tecnológica de Panamá que se muestran en el sistema. Se presenta el diagrama de caso de uso del sistema KINFO-UTP y discute cómo se combinan cuatro lenguajes de programación para hacer posible la interactividad de KINFO-UTP.

Palabras claves: interacción hombre-máquina, experiencia de usuario, nuevos paradigmas de interacción, videojuegos, motores de juego 3D, modelado 3D, animación, programación.

I. INTRODUCCIÓN

En este trabajo presentamos de forma detallada, todos los pasos que se llevaron a cabo para el desarrollo de la primera etapa del proyecto KINFO-UTP (Sistema Informativo Interactivo de la Universidad Tecnológica de Panamá-UTP), también se definen las características de KINFO-UTP que son:

- Modelado 3D del edificio # 3 del Campus Víctor Levi Sasso, de la UTP.
- Menú interactivo con el listado de todas las instalaciones físicas del edificio # 3 catalogadas.
- Escenario interactivo.
- Localización animada en 3D.

Se discute la implantación de la interface de usuario utilizando la tecnología de videojuegos, y se explica cómo se catalogaron todas las instalaciones del edificio #3 del Campus Víctor Levi Sasso, a través de encuestas y métodos de evaluación. También se muestra el diagrama de caso de uso del sistema KINFO-UTP y cómo se combinan cuatro lenguajes de programación para hacer posible la interactividad de KINFO-UTP.

La siguiente sección presenta los principales aspectos que distinguen a KINFO-UTP. En la sección III, se describe el proceso de evaluación de los motores de juego para definir con cual trabajar KINFO-UTP. A continuación, la sección IV, discute la implementación de la interface de usuario. La sección V, presenta el modelado de KINFO-UTP, describiendo el diagrama de caso de uso y la secuencia de la localización técnica. En la sección VI, se presenta el sistema de reconocimiento de KINFO-UTP.

II. KIOSCO INFORMATIVO INTERACTIVO

El sistema KINFO-UTP, en su primera etapa, tiene como función principal brindar información detallada sobre las instalaciones físicas del edificio #3, del Campus Víctor Levi Sasso, de la Universidad Tecnológica de Panamá, ya sean oficinas, aulas de clases, laboratorios y otros.

Un motor de videojuego es un término que hace referencia a una serie de rutinas de programación que permiten el diseño, la creación y la representación de un videojuego. Del mismo modo, existen motores de juego que operan tanto en consolas de videojuegos como en sistemas operativos. La funcionalidad básica de un motor de juego es proveer al desarrollador un entorno para renderizar en tiempo real gráficos 2D y 3D, un motor de física, administrador de audio, scripting, editor de animaciones, inteligencia artificial, uso de redes (cliente-servidor), streaming, administración de memoria y comunicación con diferentes dispositivos de entrada y salida [1].

UDK (Unreal Development Kit) [2] es el motor de juego utilizado en KINFO-UTP y su lenguaje de programación escrito es UScript, éste no posee una aplicación propia para programar; por ende, se evaluaron las diferentes alternativas de

III. EVALUACIÓN DE CARACTERÍSTICAS DE LOS MOTORES DE JUEGO

Basados en las características definidas para el sistema, se consideró utilizar la tecnología de videojuegos. La selección del motor de juego UDK, se fundamenta en el estudio realizado por Cata [3]. La evaluación consideró las características presentadas en la Tabla I, que compara diferentes aspectos de los motores de juego.

TABLA I. COMPARATIVA DE MOTORES GRÁFICOS PARA VIDEOJUEGOS

	Crystal Space	Quake2	V12	Unreal	Fly 3D	Genesis3D
Renderizado						
Zbuffer	Si					
Portal rendering	Si		Si	DSG	Si	Si
BSP rendering	Si	Si		Si	Si	Si
Radiosity	Si	Si		Si		Si
Stereo rendering	No					
Z-sorting	No					
Span-biffer	No					
Fogging	Si		Si	Si	Si	Si
Level of detail	Si		Si	Si	Si	Si
Halo/Corona	Si			Si		Si
Anti-aliasing	Si					
Motion blur	No					
Lens flare	Si			Si	Si	Si
Mirrors	Si			Si	Si	
Particle system	Si		Si		Si	
Landscape/Voxel (Motor/Terrenos)	Si/No		Si		Si	
Patch/curved surf	Si			Si		
Reflective surf	Si			Si		
Luces						
Colored lighting	Si	Si		Si	Si	Si
Point lighting		Si		Si		Si
Dynamic Lighting	Si	Si		Si	Si	Si
Lightmap rendering	Si		Si	Si	Si	
Phong shading	Si					
Bump mapping	Si				Si	
Gouraud-sahding	Si					No
3D sprites	Si	Si		Si		Si
APIS						
OpenGL	Si	Si	Si WIN/MAC	Throug h port	Si	No
Direct3D	Optionl port for Window	Port	Si WIN	Throug h port	Si	Si
Glide	Si					Si

software externos para utilizar UScript. Como resultado de esa evaluación se escogió Visual Studio 2008, que a través de un plugin fabricado por nFringe¹ permitía crear clases y fragmentos de código UScript.

QuickDraw 3D	No					
Texturas						
Texture mapping	Si		Si	Si	Si	Si
Persp. Texture	Si				Si	Si
MIP-mapping	Si		Si	Si		
Sub-pixel map	Si			Si	Si	
Procedural textures	Si			Si		Si
Otros						
Degrees of freedom	6					
Network support	Low-level	Si	Si	Si	Si	Low Level
Hierarchical obj	Si					Si
Collision detect	Hierarc hical boundin g box			Si		
3D sound	Si		Si	Si	Si	Si
Cartoon	No				Si	
Scene graph	No					
Language	C++			C++	C++	
Scripting language	Si (Python y LUA)	QuakeC	Si	Unreal Script	No	No
Platform	AI	AI	WIN MAC LINUX	WIN MAC	All	WIN
Source	Si		Si	Si	Si	No
Cost	LGPL	250.000 apr.(\$)	100 (\$)	500.000 apr.(\$)	li.-bre	Libre 10000 apr.(\$)
File formats	3DS, MD2 Quake, OBJ,PO V,ASE	MD2		DXF	3DS MD2 QUA KE	

Fuente: [3]

Se evaluaron además los siguientes motores de juego: **OpenSceneGraph**, es una librería gráfica libre, que permite crear visores de objetos 3D o realizar entornos interactivos con la librería de C++. Con esta librería se pueden diseñar entornos 3D interactivos, pero seria de forma muy robusta, ya que se maneja con código puro; no hay ningún editor gráfico, lo que provoca que los productos que se diseñen con esta librería demoren más tiempo en comparación a los motores de juego con interfaz grafica. **Havok**: Havok Game Dynamics SDK, es un motor físico (simulación dinámica) utilizado en videojuegos, que recrea las interacciones entre objetos y personajes del juego; por lo que detecta colisiones, gravedad, masa y velocidad en tiempo real, llegando a recrear ambientes mucho más realistas y naturales. Havok, en sus últimas versiones, se ejecuta por completo por hardware mediante el uso de la unidad de proceso gráfico (GPU, por sus siglas en inglés, Graphics Processing Unit), liberando así de dichos cálculos a la CPU. Este se apoya en las librerías de Direct3D y

² <http://epicgames.com>

OpenGL, compatibles con Shader Model 3.0. En la página web solo permiten descargar su plugin para editores 3D y un visor para ver las animaciones hechas con sus scripts, pero en sí no permiten probar en su totalidad el motor, por lo cual no se puede evaluar si es eficiente o no.

Como resultado de la evaluación se escogió UDK (Unreal Development Kit), un motor de juego gratuito, que posee todas las características necesarias para el desarrollo de KINFO-UTP; también nos permite el uso gratuito y completo del software, sin restricciones de ningún tipo, y es uno de los motores de juego más completos, además es el más utilizado en la industria de los videojuegos.

IV. INTERFACE DE USUARIO

A. Editor de Interface de Usuario (UI)

UDK poseía su propio editor de interface de usuario (UI). En el 2010 los directivos de EpicGames² removieron este editor y decidieron dejar la tecnología Scaleform, que básicamente permite colocar un diseño hecho en Adobe Flash dentro de UDK.

Lo anterior obliga a cualquier desarrollador que utilice UDK, y desea que su producto tenga una interfaz gráfica atractiva, a utilizar Adobe Flash, de otro modo deberá programar su interface en código C++. La Fig. 1 muestra la comunicación entre Flash y UDK a través de Scaleform.

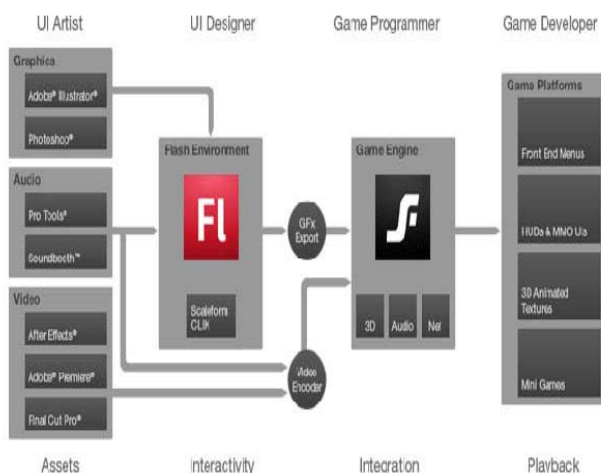


Figura N°1: comunicación entre Flash y UDK a través de Scaleform

Scaleform es un motor para renderizado de gráficos vectoriales, que se utiliza para mostrar objetos fabricados en Adobe Flash basado en interfaces de usuario. HUD (Head-up Display) y texturas animadas para juegos. Este motor fue comprado por Autodesk hace algunos años.

La implementación de esta tecnología permite crear un menú con mucha mayor calidad de diseño e interactividad

dentro de un entorno 3D. UDK posee los plugins y librerías de Scaleform gratuitas.

B. Definición Preliminar de la Interface

La primera propuesta del menú principal fue: FISC, CYT, Oficinas, Laboratorios y otros. Ver Fig. 2: primera versión del menú de KINFO-UTP.



Figura N°2: Primera versión del menú de KINFO-UTP.

Una vez que se evaluó la primera propuesta del menú, se decidió catalogar los sitios por piso; es decir, planta baja, primer piso, segundo piso, etc. Para elaborar el nuevo menú se realizó una encuesta, por correo, a usuarios potenciales, preguntándoles cómo debían estar catalogados los diferentes sitios del edificio # 3, el resultado de la misma se presenta en la Fig. 3: Menú de KINFO-UTP.



Figura N°3: Menú de KINFO-UTP

C. Menú KINFO-UTP

El menú de KINFO-UTP fue diseñado con el programa Adobe Flash utilizando el lenguaje actionscript 2. Para implementar el menú en el sistema, se realizaron los siguientes pasos:

1. Se crea una interfaz grafica en Adobe Flash de tipo actionscript 2 (la versión de UDK solo soportaba este tipo de archivo, ahora soporta actionscript 2 y 3).

² <http://epicgames.com>

2. Luego se compila en formato SWF.
3. Se importa en UDK en la carpeta flash del mismo
4. En UDK se coloca el KISMET OpenGFX.
5. Se hace referencia al SWF dentro de la librería de objetos importados en UDK.

KISMET es el otro lenguaje de programación de UDK, pero en modo gráfico. La Fig. 4, ilustra una secuencia de KISMET; como todo lenguaje, KISMET trae palabras reservadas, funciones reservadas, etc. de forma grafica. Para poder crear una función nueva dentro del KISMET se debe crear una clase de tipo KISMET en UScript y compilarla en UDK.

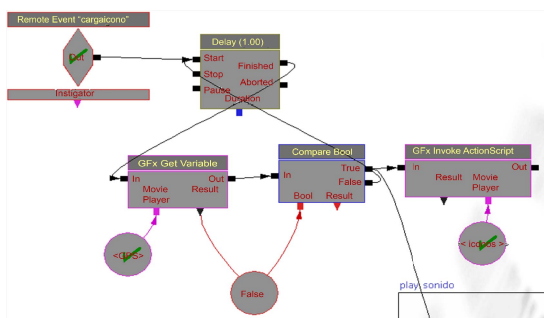


Figura N°4 Secuencia de KISMET

Este lenguaje se maneja de la misma manera que un circuito eléctrico, es decir, que trabaja por señales o pulsos eléctricos, pero en este caso son pulsos digitales.

V. MODELADO DE KINFO –UTP

A. Diagrama de Caso de Uso de KINFO-UTP

El diagrama de caso de uso que se muestra en la Fig. 5, presenta los pasos que internamente sigue el sistema una vez el usuario elige una de las instalaciones del edificio #3 del Campus Víctor Levi Sasso, de la Universidad Tecnológica de Panamá. En el punto B se explica detalladamente la secuencia de localización de KINFO-UTP tomando como ejemplo la localización de la cafetería.

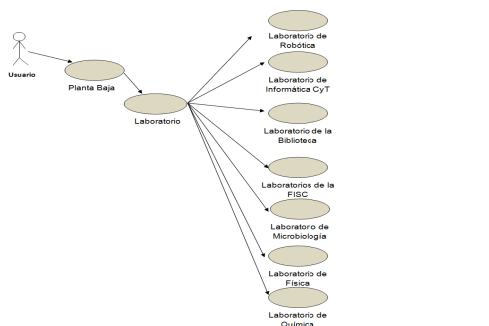


Figura N° 5 Diagrama de Caso de Uso de KINFO-UTP

B. Secuencia de localización de KINFO-UTP

A continuación presentamos la secuencia de localización técnica de KINFO-UTP:

1. El usuario selecciona un piso del menú principal (planta baja, primer piso, segundo piso, etc.)
2. Del sub-menú que se muestra en la Fig. 6 se escoge por ejemplo la opción cafetería.

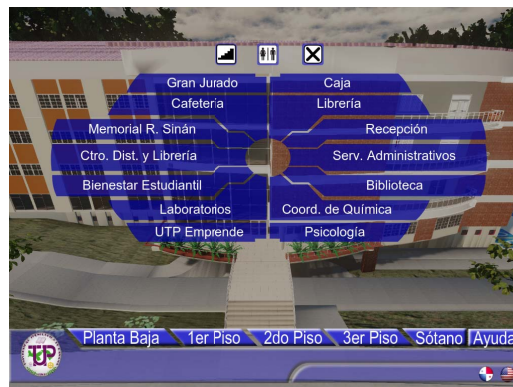


Figura N° 6: Interfaz gráfica de KINFO-UTP

3. Una vez el usuario escoge el lugar, KINFO-UTP recibe el nombre del destino y el piso al que pertenece el destino.
4. Al tener el número del piso, KINFO-UTP inicia el proceso de invisibilidad, para todo piso que esté por encima del piso al que pertenece el destino. En este caso todo piso que está arriba de la planta baja.
5. Al terminar el proceso de invisibilidad, KINFO-UTP carga la animación de la cámara para visualizar el sitio destino. Ver Fig. 7: Circuito de selección para la animación de la cámara sobre la Planta Baja.

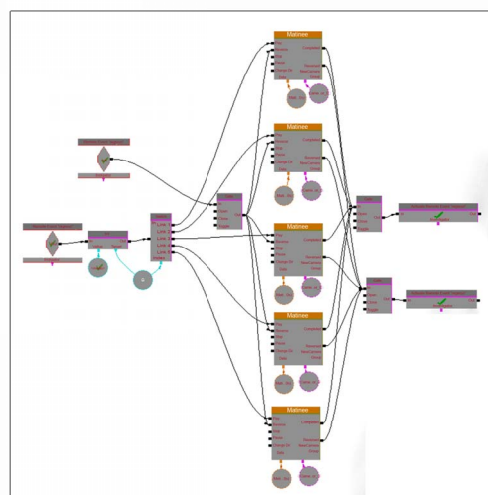


Figura N° 7: Circuito de selección para la animación de la cámara (Planta Baja).

² <http://epicgames.com>

6. Inicia la animación de la cámara.
7. Al terminar la animación se activa el SWF GPS que de forma animada escoge la vía más rápida para llegar al destino.
8. Al terminar la animación se activa el SWF íconos, que posee el pin que señalará donde está el destino. Ver Fig. 8: localización de la cafetería.
9. Al completarse la localización, el usuario podrá hacer clic en la opción regresar, si desea hacer una nueva búsqueda; si no, automáticamente después de un minuto KINFO-UTP regresará al menú.



Figura N° 8: localización de la cafetería.

VI. SISTEMA GPS DE KINFO-UTP

El sistema de posicionamiento de KINFO-UTP se basa en la colocación de hitos de referencia; es decir, se colocan puntos en una posición específica y única de acuerdo a cada sitio en el piso correspondiente, como se muestra en la Fig. 9: Hitos de la planta baja.



Figura N° 9: Hitos de la planta baja

El GPS-KINFO trabaja de la siguiente manera:

- Se colocan hitos o puntos en las esquinas del mapa de la planta baja.
- Se coloca un hito por cada puerta, no puede haber dos hitos en una misma posición, se puede usar un mismo hito para ubicar dos sitios diferentes, pero que estén uno frente al otro.
- Los hitos que coinciden en la posición, ya sea horizontal o vertical, deben tener el mismo valor en el eje de coordenadas.
- El hito inicial siempre es “las escaleras y el ascensor” llamado “h0”.
- Se debe nombrar los hitos con la secuencia h0, h1, h2, h3...
- Se permite n cantidad de hitos.
- Una vez colocados todos los hitos, se realiza un listado de caminos.
Ejemplo:
Camino #1: h0,h2,h3,h5.
Camino #2: h0,h3,h5.
Camino #3: h0,h2,h4.
- Al registrar un camino se debe empezar siempre con el hito inicial en este caso h0.
- Cuando el sistema GPS-KINFO recibe el destino a localizar, él buscará en la lista de caminos y escogerá el camino más corto.
- Cada piso posee su propio sistema GPS.
- Al escoger el camino más corto, el sistema trazará una línea, ya sea horizontal o vertical, para conectar de forma animada cada hito del camino. Ver Fig. 10: Trazado del camino para localizar la Cafetería.

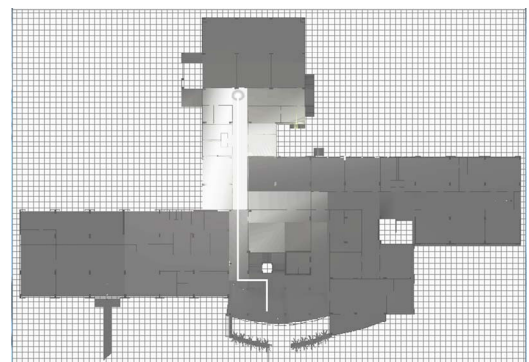


Figura N° 10: Trazado del camino para localizar la Cafetería.

VII. CONCLUSIONES Y TRABAJOS FUTUROS

El uso de motores de juego, para la creación virtual del edificio # 3, de la Universidad Tecnológica de Panamá, mediante el desarrollo del proyecto KINFO-UTP (Sistema Informativo Interactivo de la Universidad Tecnológica de Panamá), demuestra el potencial de estas herramientas para la creación de interfaces más atractivas e interactivas; así como la capacidad de interoperabilidad que las mismas ofrecen.

Entre los trabajos futuros que se esperan desarrollar para dar continuidad al proyecto presentado, podemos mencionar los siguientes:

- Incorporación de todos los edificios del campus universitario al Kiosco Informativo Interactivo (KINFO-UTP).
- Desarrollo de una versión para dispositivos móviles.
- Aplicación de realidad no inmersiva desarrollando la versión del recorrido de las instalaciones del

edificio # 3 de la Universidad Tecnológica de Panamá [4].

- Aplicación de realidad aumentada al desarrollo del proyecto.

REFERENCIAS

- [1] Motor de videojuego. Wikipedia la encyclopedia libre. www.wikipedia.com Accesado 5 de marzo de 2013
- [2] UDK –Unreal development Kit – Epic Games. <http://www.udk.com>. Accesado en abril 2011
- [3] Cata, Jordi. Comparativa de motores gráficos para video juegos. V 2.0:06-05-2002. Revisado: Alex Méndex 08-05-2002.
- [4] Nadia E. Lee G., Gisela T. de Clunie, and Gregorio Santana. 2012. 3D game engine as a visual information system. In Proceedings of the 6th Euro American Conference on Telematics and Information Systems (EATIS '12). ACM, New York, NY, USA, 331-334. DOI=10.1145/2261605.2261655 <http://doi.acm.org/10.1145/2261605.2261655>