Tecnologías de interacción avanzadas aplicadas a videojuegos

Advanced interaction technologies applied to gaming

Ing. Jonathan Donzet
Fac. Ingeniería, UdelaR, Uruguay
jonathandonzet@gmail.com

Ing. Rodrigo Leira
Fac. Ingeniería, UdelaR, Uruguay
rodrigoleira10@hotmail.com

Doc. Ing. Eduardo Fernández
Centro de Cálculo, UdelaR, Uruguay
eduardof@fing.edu.uy

Abstract

New technologies allowing new forms of interaction emerged in the last years and have been applied to videogames. From here arises the possibility of integrating these technologies in one interactive experience. In this article are presented the main results in the integration of Unity 3D graphics engine, Microsoft Kinect SDK and NVIDIA 3D Vision, in order to combine head tracking, gesture recognition and stereoscopic vision into a videogame.

Keywords: Natural user interface; stereoscopic images 3D; image processing; View-Dependent images; head tracking

Introducción

En los últimos años, varias tecnologías de interacción relacionadas con el entretenimiento han cobrado relevancia, realizando avances importantes en sus respectivas áreas y convirtiéndose en productos de consumo masivo como *Microsoft Kinect* [1] y *NVIDIA 3D Vision* [2].

Por sí solas las nuevas formas de interacción mejoran la experiencia de los usuarios, logrando mayor inmersión e interfaces más amigables. Por ejemplo el dispositivo Microsoft Kinect introduce una nueva forma de interacción de los usuarios, permitiendo el seguimiento de extremidades y tronco de múltiples usuarios, en tiempo real sobre escenarios 3D. Esta información puede ser procesada para hacer viable que el usuario interactúe de forma gestual. Por otro lado NVIDIA 3D Vision hace posible la visualización de imágenes estereoscópicas para dar sensación de profundidad a los usuarios.

En este artículo, se describe una propuesta de integración de las tecnologías antes mencionadas aplicadas al campo de videojuegos, con el objetivo de mejorar la interacción e inmersión de los usuarios. Las tecnologías utilizadas permiten realizar el seguimiento de la postura del jugador, el reconocimiento de gestos, el reconocimiento de comandos de voz, la percepción de profundidad por medio de la visualización en tiempo real de imágenes 3D estereoscópicas y la utilización de técnicas que mejoren dicha percepción.

Este artículo consta de 5 secciones. En el Estado del arte se brinda un breve resumen que describe las tecnologías utilizadas, en Desarrollo se brinda una breve descripción de los distintos elementos implementados en el prototipo, en Resultados los resultados obtenidos con el prototipo implementado, luego las Conclusiones y por último en la sección Trabajo a futuro los pasos y posibles mejoras a realizar para mejorar los resultados obtenidos.

Estado del Arte

Reconocimiento de gestos

El objetivo del reconocimiento de gestos es la interpretación de gestos humanos mediante algoritmos matemáticos. Muchos estudios se han realizado utilizando cámaras y algoritmos de visión por computador para lograr interpretar lenguajes de señas [3]. También se han realizado avances en la identificación y reconocimiento de la postura, la marcha y comportamientos humanos [4]. El reconocimiento de gestos permite interactuar directamente con computadoras, sin tener necesidad de utilizar otros dispositivos.

El Kinect es un dispositivo de detección de movimientos y captura de audio. Su principal característica es la de realizar el seguimiento corporal de los usuarios, evitando el contacto físico con un control, interpretando gestos corporales y comandos hablados [5].

Reconocimiento de voz

El reconocimiento de voz resulta muy útil para interactuar con aplicaciones sin la necesidad de tener contacto físico con ningún dispositivo, lo que puede resultar en una interacción más natural y efectiva, que complementa el uso de dispositivos como el teclado, *mouse* o controles. Estas funcionalidades se encuentran en el *Microsoft Speech Platform SDK* [6].

Generación de imágenes estereoscópicas

La estereoscopia es una técnica que logra crear la ilusión de profundidad tridimensional a partir de un par de imágenes en 2D que muestran una misma escena, generadas con un desplazamiento horizontal que las hace diferentes. Cada ojo percibe una de dichas imágenes y estas son combinadas por el cerebro e interpretadas en términos de profundidad, generando una imagen tridimensional [8,9,10].

NVIDIA 3D Vision es una tecnología desarrollada por NVIDIA que permite visualizar contenido 3D estereoscópico. El sistema se compone de un PC con una tarjeta de video, un par de gafas LCD activas y una pantalla.

Head tracking

Es una técnica que permite crear la ilusión de profundidad utilizando una pantalla 2D convencional, presentando al usuario imágenes dependientes del punto de vista.

La técnica se basa en presentar en pantalla una perspectiva de la escena que es calculada en función de la posición de la cabeza del usuario. La percepción de profundidad que se logra a través de esta técnica, se basa en la capacidad del cerebro de estimar profundidad al cambiar las posiciones relativas entre objetos, cuando el observador está en movimiento (objetos cercanos cambian de posición más rápidamente que los lejanos). Este indicador se denomina Paralaje de movimiento [11] y requiere solamente de un ojo. Junto con la estereoscopía [9] son considerados los elementos más importantes en la percepción de profundidad en distancias cortas [12].

Desarrollo

El proyecto fue dividido en dos etapas. En una etapa se estudiaron tecnologías y métodos acerca de nuevas formas de interacción, visualización estereoscópica y generación de imágenes dependientes del punto de vista, para lograr la inmersión de los usuarios en un entorno virtual. En una segunda etapa se realizó el diseño e implementación de un prototipo funcional con el objetivo de evaluar los resultados de la integración.

Los principales objetivos del proyecto fueron el diseño e implementación de una *interfaz natural* de usuario que permita la interacción con el videojuego por medio de movimientos, gestos, posturas y comandos de voz. Otro objetivo es la implementación de la técnica de *head tracking*, combinada con imágenes estereoscópicas 3D.

Integración de tecnologías

Las tecnologías involucradas en este proyecto no son compatibles directamente, por lo tanto la utilización del Microsoft Kinect SDK, Microsoft Speech Platform SDK y *NVIDIA 3D Vision* desde un motor gráfico (Unity [13]), se realizó por medio de *plugins*. Los mismos se componen de una DLL escrita en C++ que actúa como intermediaria y una clase C#, que permite invocar funciones en la DLL desde el código del videojuego (véase Figura 1). Su función general es trasladar llamadas realizadas en su interfaz, realizando invocaciones a la interfaz del SDK original.

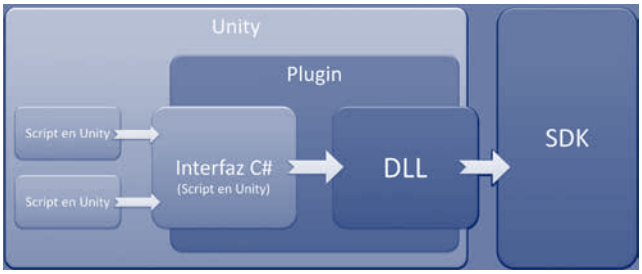

Figura 1: Arquitectura general de los plugins implementados para la utilización de los SDKs externos a Unity.

Reconocimiento de gestos

Para realizar el reconocimiento de gestos se implementó un módulo que utiliza la información de las posturas del cuerpo del usuario para determinar cuándo se efectúa cada gesto. El diseño del núcleo se compone de tres componentes:

- Gesture Part: Se encarga de reconocer un movimiento simple o postura estática.
- Gesture: Contiene un conjunto de *gesture parts* que se deben validar en secuencia. Cuando una *gesture part* no es validada, se reinicia la secuencia, de lo contrario se pasa a la siguiente *gesture part*.
- Gesture Controller: Es encargado de contener y controlar todos los *gestures*.

Existe otro tipo de gestos, que no pueden ser reconocidos con la posición de las articulaciones. Un ejemplo es reconocer si la mano está abierta o cerrada (ver Figura 2).

En la figura se observa que el estado de la mano se calcula a partir de un mapa de profundidad, del que se extraen características, que son utilizadas para clasificar entre los estados posibles (abierta, cerrada, ninguno de estos). Este proceso se enmarca dentro del reconocimiento de patrones.

Las características extraídas del mapa de profundidad son:

- Proporción de píxeles que pertenecen a la mano.
- Mediana de los histogramas horizontal y vertical.
- Desviación estándar de los histogramas.
- Cantidad de píxeles en la fila con mayor cantidad de píxeles de la mano.

Se evaluaron distintos métodos de clasificación; Redes Neuronales Artificiales [14], Prototipos [15] y Support Vector Machines [16]. Este último es el utilizado debido a que presenta mejores resultados.

A partir de este módulo se diseñó e implementó una interfaz natural de gestos para que el usuario interactúe con la Aplicación, donde sus movimientos se trasladan al videojuego.

Técnica head tracking

La técnica *head tracking* se implementó, en el videojuego, por medio de una cámara adicional en primera persona, que soporta la navegación por el escenario. En esta cámara, el *viewport* acompaña al movimiento del jugador cuando éste se traslada o rota por medio de gestos.

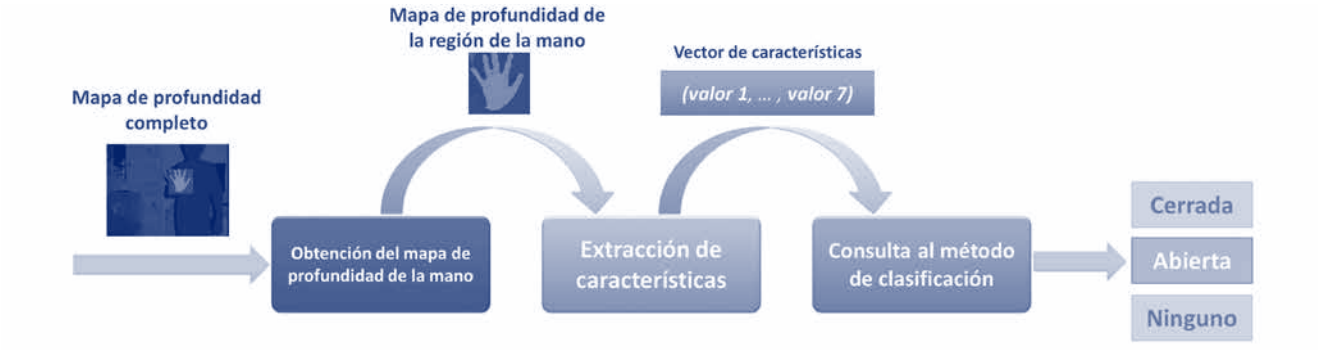

Figura 2: Etapas involucradas en el proceso de reconocimiento del estado de la mano.

Cuando el jugador se desplaza dentro de la zona de juego (ver Figura 3), el *viewport* permanece estático en el entorno virtual, mientras que el personaje se desplaza de igual manera que el jugador. Por lo tanto lo que se tiene es un mapeo de la zona de juego hacia el entorno virtual, en donde la posición relativa entre el jugador y la pantalla se mantiene para el personaje y el *viewport*.

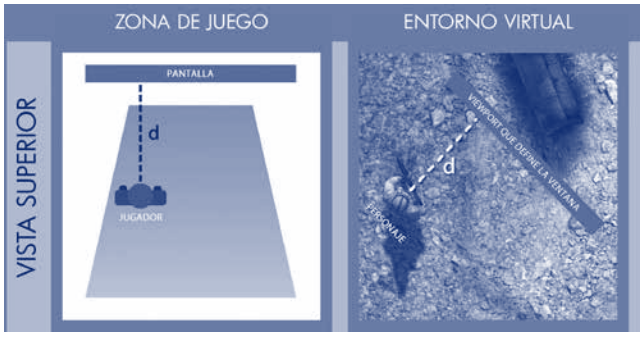

Figura 3: Vistas superiores de la zona de juego y el entorno virtual. Se muestra el mapeo realizado, donde se mantiene la misma distancia 'd' en ambos espacios.

Comúnmente, para proyecciones en perspectiva, se considera que el punto de vista se encuentra centrado respecto a la pantalla. Esto produce una pirámide truncada que se conoce como volumen de vista simétrico. Para generar una imagen dependiente del punto de vista del usuario, se debe tomar en cuenta que la posición de este es cambiante, generando un volumen de vista asimétrico [17] (véase Figura 4).

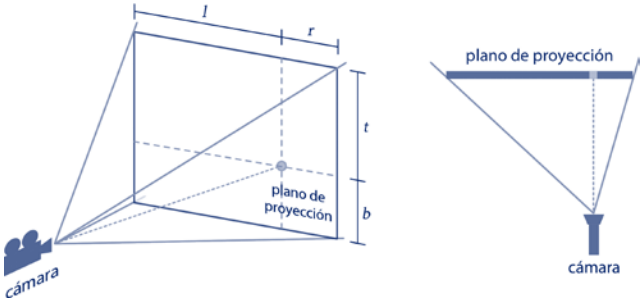

Figura 4: Volumen de vista asimétrico. La cámara determina el punto de vista del usuario en el mundo virtual.

Combinación de head tracking y estereoscopía

Si bien la técnica de *head tracking* permite crear la ilusión de profundidad utilizando una pantalla 2D convencional, resulta interesante la combinación de esta técnica con imágenes estereoscópicas, sumando los beneficios de ambos enfoques. Esto se logra ajustando en tiempo real los parámetros de ambas técnicas. De esta manera el efecto 3D se configura adecuadamente para la posición actual del jugador, con el objetivo de que las profundidades que perciba (utilizando las gafas 3D), se correspondan lo mejor posible con el punto de vista actual.

Resultados

Se desarrolló un prototipo completamente funcional, consiguiendo la integración de las distintas tecnologías involucradas a través del desarrollo de *plugins* que se conectan con el motor gráfico Unity.

Se diseñó e implementó una interfaz natural de usuario, que permite interactuar con el videojuego por medio de gestos y comandos de voz.

Se implementó un módulo de reconocimientos de gestos que incluye la apertura de la mano.

Se desarrolló la técnica *head tracking*. Se destaca la combinación de dicha técnica con imágenes estereoscópicas. Con esto se logra presentar una visualización 3D óptima, aprovechando las ventajas de ambas técnicas (estereoscopía y *head tracking*).

Finalmente, de los distintos componentes que se implementaron, existen algunos que son reutilizables. Varios de los gestos implementados podrían ser reutilizados en otras aplicaciones o videojuegos.

Resultados experimentales

El efecto de profundidad que provee la técnica headtraking depende fuertemente de dos factores. El primero es el ancho de la pantalla y el segundo es el ancho de la zona de operación del sensor, utilizado para seguimiento corporal. Se observaron mejores resultados cuando el ancho de la zona es varias veces superior al ancho de la pantalla.

Se realizó un experimento informal durante el evento "Ingeniería deMuestra 2012", en donde se puso a prueba al público un prototipo avanzado. El mismo contaba con todos los gestos implementados, *head tracking* y la integración con NVIDIA 3D Vision. Durante los dos días que duró el evento, aproximadamente

150 personas de variadas edades pudieron utilizar el prototipo. En todo momento estuvieron acompañados por al menos un integrante del proyecto el cual, además de darles instrucciones de cómo jugar, también evaluó el progreso que iban logrando en el uso de la interfaz de usuario, obteniendo opiniones de los aspectos positivos, dificultades y fallas.

Uno de los aspectos observados fue que el prototipo funcionaba mejor para algunos usuarios. Esto se debe a que varios parámetros biométricos, como la estatura y el tamaño de la palma de la mano, fueron considerados estáticos.

La retroalimentación obtenida en el evento fue de mayor importancia ya que se pudieron corregir varios de los defectos y dificultades que se detectaron.

Para evaluar la precisión de las versiones finales de los gestos implementados, se realizó un experimento simple que consistió en realizar, por un único jugador, 20 repeticiones de cada gesto, El porcentaje promedio de aciertos obtenido fue del 90%. Un experimento similar se realizó para los comandos de voz, obteniéndose un promedio de aciertos del 98%.

Para evaluar el gesto de reconocimiento del estado de la mano, se hizo un experimento formal. Se confeccionó una base de imágenes de 2800 manos a distintas distancias y posiciones. El 80% de la base se utilizó para el entrenamiento del método de clasificación, el 20% restante se utilizó para su validación. El porcentaje promedio de aciertos obtenido fue del 90%.

La utilización de gestos y comandos de voz tiene algunas desventajas frente a otras formas de interacción. Por ejemplo cuentan con falsos positivos (detección errónea de una acción del usuario, que no representa su intención real) a la vez que su utilización tiende a fatigar con mayor intensidad. Para ciertas aplicaciones la interacción con teclado y *mouse* es sensiblemente más eficiente y eficaz.

Conclusiones

El trabajo realizado es un acercamiento práctico a los temas tratados. A lo largo del proyecto se obtuvo conocimiento y experiencia en la combinación de tecnologías, para la implementación de soluciones de esta índole.

La mayoría del trabajo realizado puede ser aprovechado para el desarrollo de aplicaciones interactivas fuertemente basadas en gráficos 3D, por ejemplo en visualización arquitectónica [18].

Trabajo futuro

Realizar experimentos formales de la interfaz natural de usuario, guiadas por un test de usabilidad.

A partir de la información percibida del usuario seleccionar automáticamente los parámetros biométricos.

Con los módulos reutilizables que se desarrollaron y con las tecnologías ya integradas, implementar un *framework* para facilitar el desarrollo de aplicaciones basadas en interacción por gestos y comandos de voz, imágenes estereoscópicas y *head tracking*.

Referencias

[1] Kinect for Windows, Microsoft. Retrieved from http://kinectforwindows.org

[2] NVIDIA 3D Vision. Retrieved from http://www.nvidia.com/object/3d-vision-system-requirements.html

[3] Jayashree R. Pansare, Shravan H. Gawande, Maya Ingle (2012). Real-Time Static Hand Gesture Recognition for AmericanSign Language (ASL) in Complex Background. Journal of Signal and Information Processing, 2012, 3, 364-367 doi:10.4236/jsip.2012.33047 Published Online August 2012

[4] Vladimir I. Pavlovic, Rajeev Sharma and Thomas S. Huang. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. Retrieved from http://www.cs.rutgers.edu/~vladimir/pub/pavlovic97pami.pdf

[5] Kinect for Windows features, Microsoft. Retrieved from http://www.microsoft.com/en-us/kinectforwindows/discover/features.aspx

[7] Microsoft Speech Platform SDK, MSDN, Microsoft. Retrieved from http://msdn.microsoft.com/en-us/library/hh361572(-v=office.14).aspx

[8] Simanek, R. (2002). The Illusion of Reality in Stereoscopy. Retrieved from http://www.lhup.edu/~dsimanek/3d/stereo/reality.htm

[9] Steinman, S., Steinman, B., Garzia, R. (2000). Foundations of Binocular Vision: A Clinical perspective, McGraw-Hill Medical. ISBN 0-8385-2670-5.

[10] Calvert, J. (2005). Optics and Visual Perception. Retrieved from http://mysite.du.edu/~jcalvert/optics/stereops.htm

[11] Irvin, R. (1985). La percepción. Barcelona, ES, Prensa científica. ISBN 8475930190.

[12] Cutting, J., Vishton, P. (1995). Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth. San Diego, CA, Academic Press. Retrieved from http://pmvish.people.wm.edu/cutting&vishton1995.pdf

[13] Unity engine. Retrieved from http://unity3d.com

[14] Redes Neuronales Artificiales, Curso de Métodos de Aprendizaje Automático, Facultad de Ingeniería, Universidad de la República – Uruguay. Retrived from http://www.fing.edu.uy/inco/cursos/aprendaut/teorico/4-redes.pdf

[15] Alba, J., Cid, J. (2006). Reconocimiento de patrones. Carlos III, Madrid, Universidad Carlos III, Madrid. Retrived from http://www.gts.tsc.uvigo.es/pi/Reconocimiento.pdf

[16] Souza, C. (2010). Kernel Support Vector Machines for Classification and Regression in C#. Retrived from http://crsouza.blogspot.com/2010/04/kernel-support-vector-machines-for.html

[17] Kooima, R. (2009). Generalized Perspective Projection. Louisiana, US, Louisiana State University. Retrieved from http://csc.lsu.edu/~kooima/pdfs/gen-perspective.pdf

[18] Winscape, Rational craft. Retrieved from http://www.rationalcraft.com/Winscape.html