

Propuesta de Trabajo Final
Carrera de Ingeniería de Sistemas

Entorno para el desarrollo de sistemas interactivos y su aplicación al marketing digital

Alumnos:

Santiago Carliski

Maximiliano Luis Cassola

Director:

Dr. Juan Pablo D'Amato

Codirector:

Dr. Cristian García Bauza

1. Introducción

En los últimos años, los medios de interacción con la computadora han evolucionado rápidamente gracias a la aparición de nuevos dispositivos de adquisición de información. A partir de la década del 80 y hasta ahora, el usuario interactuaba mayormente utilizando únicamente teclado y mouse a través de interfaces gráficas. En los últimos años ha surgido otra familia de medios donde el usuario utiliza su propio cuerpo a través de *gestos* para comandar la computadora, concepto conocido como Sistemas de Interfaz Natural (NUI, por las siglas en inglés de *Natural User Interface*) [1] [3] [4] [5].

La industria del videojuego ha promovido la divulgación de tecnologías NUI que sólo se encontraban dentro del ámbito científico o la cinematografía. Entre los nuevos dispositivos, se puede nombrar a Surface [6], WiiMote [7], PS@Move [8] y particularmente Kinect [2]; entre otros. Este último es el que más ha revolucionado el mercado (se vendieron 10 millones de unidades en los primeros tres meses de venta [9]) y fue su aparición la que disparó en mayor medida el desarrollo de aplicaciones con interfaz natural que no están relacionadas con los videojuegos [10].

2. Motivación

El sensor Kinect cuenta con un sensor de profundidad y una cámara. Esto permite capturar el movimiento 3D del cuerpo en tiempo real (30 imágenes por segundo), simplificando enormemente el reconocimiento de las partes del cuerpo y al mismo tiempo permite detectar el gesto que el usuario está realizando. Un área que se puede beneficiar con el uso de esta

tecnología novedosa es la del *marketing*. Gracias al potencial de innovación y bajo costo del dispositivo, se lo puede pensar como una nueva herramienta de promoción para atraer público.

El mayor desafío al momento de utilizar este tipo de tecnologías es contar con una forma de control estandarizado. Cada desarrollo propone una forma distinta de operación, por lo que los programadores deben entender la utilización del dispositivo dependiendo del contexto de aplicación. Al mismo tiempo, las aplicaciones a desarrollar deben resultar atractivas, por lo cual se utilizan entornos de desarrollo orientado a diseñadores gráficos como Flash o HTML5. Lamentablemente, estos productos no tienen una forma nativa de integrar estos nuevos dispositivos.

Adicionalmente, otro de los desafíos que se presentan a la hora de poner en práctica estas nuevas tecnologías es la heterogeneidad de controladores o *drivers*, lo que lo hace dificultoso para unificar diferentes aplicaciones. Existen dos líneas principales de grupos desarrollando *drivers* para Kinect. Microsoft, que cuenta con los *drivers* oficiales [11] para el sensor, y por parte del software libre se pueden utilizar los provistos por OpenNI [12]. No son los únicos existentes pero son los proyectos con mayor actividad y progreso en la actualidad. Si bien ambos tienen como objetivo principal la detección del cuerpo humano, presentan diferencias en las estructuras de datos por lo que se vuelve más complejo cambiar de plataforma para una misma aplicación.

Una de las propuestas de esta tesis es abstraer el comportamiento común entre ambos *drivers* y permitir, a manera de *wrapper* utilizar Kinect, indistintamente del SDK elegido. Para esto es necesario unificar las estructuras de datos y la forma de comunicación entre las aplicaciones y el dispositivo. Si bien existen algunas propuestas similares, las mismas están pensadas para lenguajes de desarrollo específicos y aquí se espera que se pueda extender de forma genérica a distintos lenguajes de programación (C#, C++, Java, etc.) e incluso plataformas para desarrollos 3D tales como Unity3D. A su vez, se pretende incluir otras herramientas, tal como filtros de distancias o selección de zonas de interés, que faciliten la tarea del programador.

3. Objetivos

El objetivo principal de este trabajo consiste en el desarrollo de un entorno o plataforma que actúe como proveedor de datos para aplicaciones que hacen uso de NUI, utilizando una arquitectura Cliente-Servidor. El cliente, que puede estar desarrollado en diferentes lenguajes, debe contar con una interfaz de métodos y funciones de inicialización, acceso y comunicación. El servidor publica dichos métodos y cumple el rol de abstracción de las diferentes representaciones internas, tal como imágenes RGB y profundidad, esqueletos y gestos, permitiendo así una representación general y un uso más claro e intuitivo. A su vez, permite intercambiar entre los dos principales *drivers* que dan soporte al sensor de movimientos Microsoft Kinect. Esto toma mayor importancia si se piensa en la posibilidad de utilizar Kinect en plataformas de desarrollo específicas para diseño de interfaces gráficas o entornos 3D.

Para la comunicación entre servidor y cliente se planteará un protocolo de comunicación diseñado específicamente para este tipo de tecnología, teniendo en cuenta que los paquetes que viajan entre servidor y cliente no son críticos. Dentro de cada paquete o trama, se codificará información sobre los esqueletos, gestos detectados y su imagen correspondiente.

Para la detección de gestos se analizarán algunas estrategias conocidas en la literatura, atendiendo a los gestos más comunes tal como swipe, zoom, panning entre otros. Al mismo tiempo, se espera poder extraer y registrar información que pueda llegar a servir en posibles análisis, tal como la duración y velocidad de un gesto, datos biométricos como la altura de los individuos, mínima distancia de captura, entre otras y también consultar el estado del dispositivo en cada momento. Finalmente se propondrán ejemplos en distintos lenguajes utilizando esta plataforma.

4. Cronograma de Actividades

A continuación se enumeran las actividades a realizar:

1. Relevamiento bibliográfico. 1 mes.
2. Análisis de los desarrollos disponibles similares. 1 mes.
3. Diseño y desarrollo del protocolo de comunicación. 2 meses.
4. Diseño y desarrollo de los distintos módulos. 2 meses.
5. Implementación de aplicaciones de ejemplo. 1 mes.
6. Evaluación de las herramientas generadas. 1 mes. (*)
7. Documentación e informe final sobre el desarrollo. 2 meses. (*)

(*) Se realizará en paralelo con el resto de las actividades.

5. Referencias

[1] Carroll, John M. The Encyclopedia of Human-Computer Interaction, 2nd Ed.. ISBN: 978-87-92964-00-7. The Interaction Design Foundation. (2009)

[2] Microsoft Kinect. URL: <http://www.kinectforwindows.org> (2013)

[3] Mohd Kufaisal bin Mohd Sidik, Mohd Shahrizal bin Sunar, Ismahafezi bin Ismail, Mohd Khalid bin Mokhtar, Normal binti Mat Jusoh. Paper: A Study on Natural Interaction for Human Body Motion using Depth Image Data. Workshop on Digital Media and Digital Content Management. (2011)

[4] Petersen, Nils, Stricker, Didier. Paper: Continuous Natural User Interface: Reducing the Gap Between Real and Digital World. IEEE International Symposium on Mixed and Augmented Reality. Science and Technology Proceedings. 978-1-4244-5419-8/09. (2009)

[5] Wigdor, Daniel, Wixon Dennis. Brave Nui World: Designing Natural User Interfaces for Touch and Gesture. Estados Unidos. Morgan Kaufmann Publishers. ISBN: 978-0-12-382231-4. (2011)

[6] Microsoft Surface. URL: <http://www.microsoft.com/surface/en-us> (2013)

[7] Nintendo Wii. URL: <http://www.nintendo.com/wii> (2013)

[8] Play Station Move. URL: <http://us.playstation.com/ps3/playstation-move/> (2013)

[9] Zeng, Wenjun. Microsoft Kinect Sensor and its effects. (2013)

[10] Abhijit J. Kinect for Windows SDK Programming Guide: Build motionsensing applications with Microsoft's Kinect for Windows SDK quickly and easily. Packt Publishing. (2012)

[11] Microsoft Kinect SDK. URL: <http://www.microsoft.com/en-us/kinectforwindows/develop/overview.aspx> (2013)

[12] OpenNI. URL: <http://www.OpenNI.org> (2013)