

Animación por Computador y Videojuegos

Space Invaders con Kinect

Jordi Mansanet Sandín

18 de enero de 2012

1. Introducción

Este trabajo consiste en modificar el juego *Space Invaders* partiendo del código fuente suministrado por el profesor. Cada alumno debe modificar dicho código dependiendo del trabajo asignado, por lo que se ha utilizado un sistema de versiones basado en *Subversion*. Gracias al programa **Totoise SVN** se puede acceder y modificar fácilmente los archivos del repositorio de las versiones.

En mi caso se ha adaptado el manejo de la nave para poder moverla a través de movimientos de la mano. Estos movimientos son capturados con el dispositivo Kinect creado por **PrimeSense** para **Microsoft**. Gracias al *framework* **OpenNI** [1] podemos comunicarnos fácilmente con la información que nos proporciona el sensor Kinect. Por otro lado, con el *middleware* **NITE** [2] [3] tenemos acceso a ejemplos que analizan dicha información y la usan para detectar determinados movimientos con partes del cuerpo. En la Figura 1 vemos el proceso de adquisición y procesado de la información para producir una aplicación final.

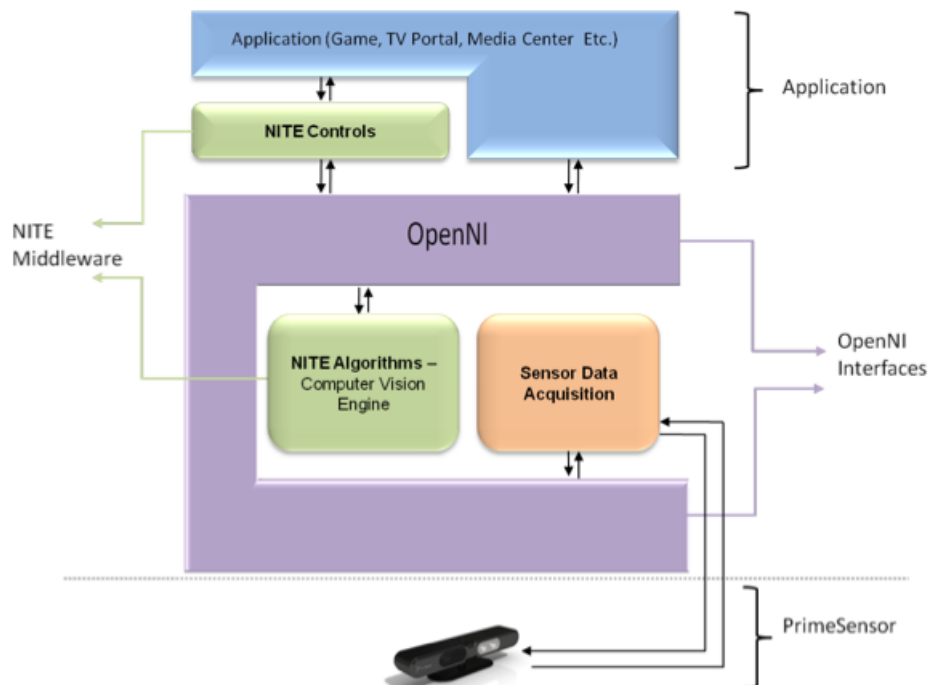


Figura 1: Adquisición y procesado con Kinect

2. Metodología

Para poder ejecutar el programa necesitamos tener instaladas las librerías **NITE** y **OpenNI**, las cuales nos proporcionan las funciones necesarias así como los *drivers* para conectar con la Kinect a través del USB.

Para que el programa sea compatible tanto para jugar con Kinect como con teclado se ha decidido crear una nueva configuración de compilación en el **Visual Studio** llamada **Debug_Kinect**, en la cual se incluyen todas las librerías y rutas necesarias. Además esta configuración tiene una variable definida en el preprocesado (Propiedades de configuración → C/C++ → Preprocesador) llamada **USE_KINECT**. Con ella podemos decidir cuándo se compila el código referido a la Kinect insertándolo en una sentencia tipo **#ifdef USE_KINECT ... #endif**.

Para realizar el trabajo nos hemos basado en los ejemplos que se nos proporciona al instalar **NITE**, en concreto **SingleControl** y **Boxes**. Gracias a **NITE** existen unos objetos llamados *Point Controls*, los cuales reciben información sobre los puntos de la mano en cada frame (mediante *callbacks*), la analizan y realizan una determinada acción. En nuestro caso hemos utilizado el objeto *SelectableSlider1D*. Éste reconoce el movimiento de la mano a lo largo de uno de los tres ejes (X, Y, Z) y devuelve un valor entre 0 y 1 según la posición relativa a un *slider*. Como nuestra nave tiene tres posibles movimientos (izquierda, centro y parada) hemos dividido el *slider* en 3 zonas. Dependiendo de donde caiga el valor devuelto (posición de la mano) la nave se moverá hacia un lado u otro, o permanecerá parada.

Centrándonos en el código, por un lado hemos creado un archivo **kinect.cpp/.h** donde metemos todas las funciones referentes a la Kinect. Podemos ver cómo están implementados los posibles *callbacks*, así como algunas funciones que se llamarán desde otros archivos comunes del juego. Se ha decidido hacer la inicialización de la Kinect justo después de la carga del juego (estado **CG_FADING_IN** en el fichero **si_Basecode.cpp**). En ese momento se ejecuta la función **initKinect()**, la cual arranca la kinect y la deja a la espera de alguna acción.

Por otro lado, el objeto *SelectableSlider1D* comienza a funcionar cuando detecta un gesto concreto, llamado *Wave* en la literatura. Éste consiste en mover la mano de izquierda a derecha de forma continuada y con la palma mirando el sensor. Esta detección la realiza la función **waitKinect()**, la cual se ejecuta cuando el juego está en el estado **CG_WAIT_KEY**. De esta forma podemos empezar a jugar tanto realizando el gesto antes descrito, o apretando cualquier tecla como dice la pantalla.

Finalmente la función **updateKinect()** obtiene un valor 0, 1 o 2 (izquierda, parado o derecha) dependiendo de la posición del *slider*. Una vez se

tiene el movimiento deseado se llama a la función ya implementa **Update()**, en la que se maneja el evento de mover la nave hacia un lado o hacia otro. La variable global **g_dir_kinect** almacena el movimiento que debe tomar la nave.

Por último destacar que se han añadido los ficheros **kbhit.cpp/.h** y **signal_catch.cpp/.h** debido a que en los ejemplos de **NITE** eran necesarios.

3. Conclusiones

Con este trabajo hemos podido tener una visión general de cómo está implementado el juego *Space Invaders* y cómo se puede adaptarlo para utilizar otros dispositivos externos como la Kinect. Gracias a los algoritmos y ejemplo dados por **NITE** no es demasiado complicado incluir la utilización de sensores como la Kinect en tus aplicaciones.

Dado que este trabajo ha sido una primera versión sobre el tema, en el futuro se podrían realizar otros trabajos sobre ello. Por ejemplo: añadir la posibilidad en el menú principal de utilizar o no la Kinect, inclusión de otro tipo de gestos para otras tareas, depurar y optimizar el código, etc.

Referencias

- [1] *OpenNI User Guide*.
- [2] *Prime Sensor NITE 1.3 Algorithms Notes*.
- [3] *PrimeSense - NITE Controls User Guide*.