

PSMove MIDI Drum

Alexander Pinto De la Gala
Universidad Católica San Pablo
Departamento de Ciencia de la Computación
{alexander.pinto@ucsp.edu.pe}

Abstract

Various motion capture devices and sensors have been used for the interpretation of virtual musical instruments. Introducing PSMove MIDI Drum, a virtual drum kit that uses the benefits of the PSMove controller and the PSEye camera to simulate the performance of a real drum kit. The application meets the proposed objectives having a good general behavior. Some inconveniences will be realized in the capture of events through collisions which were not successful with fast movements.

Resumen - Diversos dispositivos de captura de movimiento y sensores han sido utilizados para la interpretación de instrumentos musicales virtuales. Presentamos PSMove MIDI Drum, una batería de percusión virtual que utiliza las bondades del controlador PSMove y la cámara PSEye para simular la ejecución de una batería real. La aplicación cumple con los objetivos propuestos teniendo un buen comportamiento general. Algunos inconvenientes se dieron en la captura de eventos por medio de colisiones las cuales no fueron exitosas cuando el movimiento es rápido.

I. INTRODUCCIÓN

Diversos dispositivos de captura de movimiento han sido utilizados para la interpretación de instrumentos musicales virtuales, a través de los gestos y posicionamiento propios de un ejecutante, es posible simular el desempeño de un músico [1].

En el año 2010, Sony presentó el dispositivo PlayStation Move Motion Controller (PSMove), el cual puede ser emparejado con una computadora por medio de diferentes APIs y servicios. Estos dispositivos sumados al uso de una cámara puede ser utilizado como dispositivo de seguimiento (*tracker*). Si bien existe herramienta del propietario para el manejo de PSMove, estas solo pueden ser utilizadas bajo la plataforma de PlayStation. Perl [2], presenta una primer biblioteca (API) libre multiplataforma para PSMove. Posteriormente ampliada en forma de servicio por [3] denominada PSMoveService.

PSMove MIDI Drum, es una aplicación que utiliza dispositivos PSMove para simular la ejecución de una batería de tambores y percusión (*drum*). La aplicación es desarrollada con el motor Unity, de la cual es aprovechada sus funcionalidades para la detección de colisiones, las cuales generan eventos (*triggers*) que son transformados a su vez, dependiendo del instrumento virtual colisionado, en eventos MIDI. Para la generación de los eventos MIDI se utilizó la biblioteca NAudio [4].

El desarrollo de esta aplicación es parte del Proyecto Final del Curso Interacción Humano Computador.

II. MOTIVACIÓN

Si bien existen experiencias similares para la ejecución de instrumentos musicales virtuales, el dispositivo PSMove ha tenido poca atención para este objetivo, por lo tanto el objetivo directo es la experimentación de PSMove para entender su funcionamiento, conocer sus capacidades y limitaciones como herramienta de ejecución de instrumentos virtuales, en este caso una batería de percusión. Como objetivos secundarios tenemos la identificación de conceptos de técnicas de interacción humano computador como navegación, selección, manipulación y control. Finalmente dar versatilidad a la aplicación por medio de un lenguaje universal de comunicación musical como es MIDI, lo que permitiría que la experiencia pueda ser extrapolada para otros instrumentos musicales.

III. TRABAJOS RELACIONADOS

El uso de sensores y tracking para la ejecución de instrumentos virtuales se da en diferentes trabajos. Han [1] en el 2014, utiliza el dispositivo LeapMotion para simular un piano, teniendo buenos resultados.

En cuanto al uso de PSMove, el trabajo de Perl es el de mayor influencia [2], y es base para otros proyectos como PSMove Service [3] el cual es utilizado para la implementación de la presente.

Una experiencia similar es el software comercial Aerodrum [5], el cual se basa en visión computacional.

IV. PROPUESTA DE PROYECTO

El Proyecto tiene como objetivo implementar una aplicación de simulación de ejecución de un instrumento de percusión (batería de tambores), con el uso de los dispositivos inalámbricos PSMove y cámara PSEye, los cuales harán el seguimiento (tracking) de los movimientos usuales de un músico baterista. Los movimientos capturados permitirán calcular colisiones con el instrumento virtual, estas colisiones deben generar eventos que serán traducidos a un lenguaje de comunicación de instrumentos digitales (MIDI). El mensaje MIDI tendrá información acerca del sub instrumento ejecutado por medio de key (nota) y de la

intensidad del golpe (timbre/volumen) que se conoce como velocidad en el lenguaje MIDI. Los mensajes MIDI podrán ser recibidos desde cualquier instrumento MIDI existente.

En el ambiente 3D de la aplicación se encuentran ubicados espacialmente los sub instrumentos de la batería (bass drum, snare, hi-hat, ride). Los dispositivos PSMove controlan la posición así como la rotación de las baquetas (drumstick), se ha modelado los drumstick de manera que su malla de colisión sólo se encuentra en unos de sus extremos, coincidiendo con la lado del globo luminoso de los PSMove. En esta primera etapa se ha utilizado dos mandos PSMove correspondiendo cada uno a una baqueta.

Se ha simplificado la malla de colisión de todos los instrumentos, para mejorar el desempeño de la aplicación en cuanto a colisiones. Las colisiones serán del tipo continuo mejorando en cierta medida la velocidad de respuesta. Otra mejora en las colisiones es por medio de código el cual solo se produce cuando exista un movimiento hacia abajo evitando golpes que no se darían en un ambiente real.

Para mejorar la interacción con el usuario se da una retroalimentación (feedback) de las colisiones por medio del vibrador incorporado de los PSMove.

V. TECNOLOGÍAS USADAS

A. PlayStation Move Motion Controller

El dispositivo PSMove es un controlador de juegos cilíndrico con esfera de color blanco en uno de sus extremos. La esfera se ilumina desde su interior por un led RGB, permitiendo diferentes colores. El controlador contiene 9 botones: cuatro botones en su parte superior redondos marcados como cuadrado, triángulo, cruz y círculo; dos botones a sus costados marcados como select y start, un botón grande central marcado con el logo de PSMove, un pequeño botón con el logo de PlayStation utilizado para encender y apagar el mando, y un disparador analógico marcado con la letra T. Todos los botones son analógicos, excepto el trigger.

El mando también posee puertos de conexión: un socket mini USB, para su pareo y carga; conectores para su carga en un puerto del fabricante y un puerto de expansión para accesorios adicionales.

El control cuenta con los siguientes sensores: acelerómetro, giroscopio y magnetómetro. Se indica que también cuenta con un sensor de temperatura el cual no es utilizado por la API. El acelerómetro y el giroscopio envían valores de 16 bits, el magnetómetro de 12 bits.

El PSMove envía hasta 85 actualizaciones por segundo via bluetooth.

Graf. 1: Controladores PlayStation Move



B. Cámara PsEye

La cámara PSEye es de bajo costo, tiene conectividad USB 2.0 y un frame rate de 60 FPS a 640x480 y 120 FPS a 320x240. Permite además cambiar valores como duración de la exposición, lo cual es útil para evitar problemas en el tracking.

Graf. 2: Cámara PlayStation Eye



C. Herramientas y Software de desarrollo

La aplicación además utiliza diferentes herramientas para su desarrollo. La captura de posición de los dispositivos PSMove, así como el uso de sus botones y vibración; se utiliza el API PSMove Service, el cual está implementado en C++. La interfaz gráfica (GUI), el ambiente gráfico 3D, están implementadas bajo la IDE Unity la cual trabaja con el lenguaje C#. Debido a la incompatibilidad directa de estos dos lenguajes es necesario crear un Wrapper y una librería dinámica entre PSMoveService y Unity, para la integración de los dispositivos dentro del ambiente de trabajo 3D.

Para la generación de mensajes MIDI se utiliza la biblioteca NAudio. Las colisiones generan diferentes mensajes MIDI dependiendo del instrumento, lo cual significa una nota distinta, además de la fuerza del impacto. Para la simulación de la fuerza del impacto se utiliza el cálculo de velocidad rotacional que calcula PsMoveService, y es transformada en términos de velocidad para MIDI con valores de 0 a 127, donde 0 es el valor mínimo el cual equivale a silencio.

Las pruebas de recepción de los mensajes MIDI se hicieron con el uso del software BFD3, además de una aplicación de direccionamiento de mensajes, loopMIDI.

VI. IDENTIFICACIÓN DE EVENTOS DE SELECCIÓN, NAVEGACIÓN Y MANIPULACIÓN

A. Navegación

Los eventos de navegación se producen por el uso de los dispositivos PSMove, con estos el usuario puede "navegar" el ambiente 3D, por medio de las baquetas, así como recorrer entre las opciones del ambiente 2D de la ventana inicial. Esta navegación se realiza por el movimiento de los PSMove y el seguimiento conjunto de la cámara PSEye. Para el caso del ambiente de Calibración, la navegación en cierta forma es guiada por medio de la visualización de flechas que indican donde es la posición del instrumento "tarola", el cual permite la ubicación inicial para reconocer el resto de instrumentos.

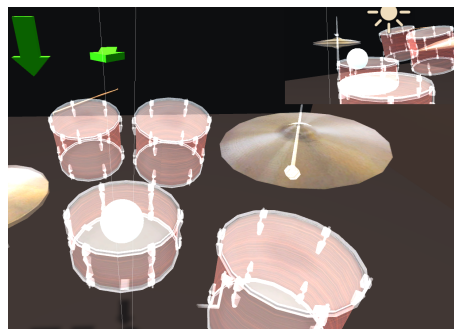
Graf. 3: Pantalla Inicial



B. Selección

La selección se presenta en dos casos, el primero es en la ventana inicial, la cual es un ambiente 2D, en donde el usuario puede elegir entre las opciones de jugar o calibrar. Ubicado sobre los botones de opciones puede confirmar su selección por medio del botón X del mando. En el ambiente 3D, las selecciones son de cierta forma de manera más automática. Luego que la "pre-selección" del instrumento se haya dado (ubicando las baquetas sobre el instrumento), se procede a confirmar su selección por medio de un gesto similar al de un músico baterista, con un movimiento de arriba hacia abajo, y preferentemente de manera rotacional con el uso de las muñecas.

Graf. 4: Pantalla Calibración



C. Manipulación

De la misma forma que la selección es intuitiva en el ambiente 3D, lo es la manipulación. El movimiento de las baquetas es apoyado por el uso del acelerómetro, giroscopio y magnetómetro de los PSMove, esto permite que las baquetas virtuales tengan un comportamiento similar al real, pudiendo girarlas, variar su orientación y rotarlas. Luego que se ha seleccionado el instrumento (simulando el golpe real de un tambor), se puede a la vez manipular esta selección, es decir el golpe puede tener una intensidad variable, de acuerdo a la velocidad de giro del mando, simulando golpes débiles y fuertes.

Graf. 5: Pantalla Juego



VII. PRUEBAS Y RESULTADOS

Las pruebas realizadas se dieron por medio del uso de la aplicación. Las Pruebas de tracking fueron satisfactorias siempre y cuando se tuviera en cuenta la abertura de la cámara PSEye. por lo que debieron de corregirse algunos valores para que el movimiento sea el adecuado. Las pruebas para el uso de botones fueron buenas, se pudieron utilizar para la selección de las opciones en la pantalla de bienvenida. Las prueba del uso del vibrador incorporado fueron buenas, pero se notó un cierto retardo en el inicio, que podría provocar un poco de confusión. Las pruebas en colisiones no fueron del todo exitosas, en movimientos lentos y pausados se pudo obtener respuestas acertadas, sin embargo cuando los movimientos eran rápidos y continuos las colisiones no fueron detectadas por lo que el instrumento no tuvo el comportamiento deseado.

VIII. CONCLUSIONES

El dispositivo PSMove y la PSEye es un sensor y que ofrece buenos resultados teniendo en cuenta su bajo costo y el ser una tecnología con más de diez años. La precisión en la captura de los movimientos en los ejes x y y es muy buena, sin embargo en profundidad no es tan acertada. El principal problema de la aplicación es en la detección de colisiones, la cual no permite una operación continua y sin fallos, este problema está relacionada más con el motor Unity.

REFERENCES

- [1] J. Han and N. Gold, "Lessons learned in exploring the leap motion™ sensor for gesture-based instrument design." Goldsmiths University of London, 2014.
- [2] T. Perl, B. Venditti, and H. Kaufmann, "Ps move api: a cross-platform 6dof tracking framework," in *Proceedings of the Workshop on Off-The-Shelf Virtual Reality*, 2013, p. 8.
- [3] Boulay. [Online]. Available: <https://github.com/psmoveservice/PSMoveService>
- [4] M. Heath. [Online]. Available: <https://github.com/naudio/NAudio>
- [5] Aerodrum. [Online]. Available: <https://aerodrums.com/home/>