

# Developing an Open Source Exertion Interface for Two-Handed 3D and 6DOF Motion Tracking and Visualisation

Jennifer G. Sheridan  
BigDog Interactive Ltd.  
London, UK  
[jennifer@bigdoginteractive.com](mailto:jennifer@bigdoginteractive.com)

**Novel technologies offer the potential for tracking and visualizing whole body movement in new ways which opens up possibilities for creating new forms of interaction. We highlight the problems and opportunities for designing for and the visualising of six degree of freedom (6DOF) motion tracking for absolute, two-handed input using the Nintendo Wiimote as our baseline platform and a playground clapping game as our context. We present a new technique for combining linear movement, rotation and vision tracking for two-handed motion tracking and provide links to open source tools and applications for next generation 6DOF motion tracking and visualisation of exertion games.**

*6DOF, degrees of freedom, 3D, Wiiremote, playground games, clapping, absolute positioning, DIY, Opie, Physics.*

## 1. INTRODUCTION

Novel technologies offer the potential for tracking and visualizing whole body movement and interaction [England et. al, 2009] in both three-dimensional space (3D) and in 6 degrees of freedom (6DOF). These technologies open up possibilities for creating new forms of interaction. Most recently, computer game controllers and technologies such as the Nintendo Wiimote [Nintendo], Sony Playstation Move [Sony] and the Microsoft Kinect [Microsoft] present novel methods for capturing expressive gesture and motion tracking. For example, Microsoft Kinect allows participants to engage with the games through “controller-less” technology – participants simply wave their hands in the air to issue commands and to control the game.

Open source development environments and applications provide potential solutions for creating low-fi, 6DOF systems and games. As large corporations begin to embrace open source solutions, we are beginning to see a plethora of innovative design thinking and practice [Paradiso et, al, 2008]. However, little documentation exists which fully explains the general hardware and software constraints that underlie the open source development of 6DOF motion tracking. As such, how these systems are developed and how the HCI

community could benefit from this knowledge has largely been under explored.

We seek to demystify the underlying principles that surround development of 6DOF tracking and to illustrate example prototype solutions using open source tools. Our aim is to provide designers with low-fi and low-cost prototyping tools and applications for designing for 6DOF.

In this paper we present a case study for designing and visualising 6DOF and highlight the problems and opportunities using the Nintendo Wiimote as a baseline platform. Our challenge is to develop the system and application *without* using the Nintendo Wii development kit or Nintendo Wii gesture-tracking library. Our aim is to create an open source application and system using open source tools only. Our intention is to illustrate how iterative DIY development can form a core part of HCI research practice.

While many studies have explored using a Wiimote as an input device for interaction [Bradshaw et. al, 2008; Benovoy, M., 2008; Lee, 2008], little information is available about the constraints that Wiimotes pose when used for *absolute* 6DOF motion capture for two-handed input. We discuss how this knowledge is useful for both HCI designers and researchers alike using children’s playground clapping games as our context.

First, we describe 6DOF and the technical solutions for using a Wii. We chart the iterative design and development of an open source 6DOF motion-tracking device using a combination of the Wii and a vision-tracking solution. We discuss our exploratory usability testing and finally, summarise our lessons learned. This paper provides designers with open source solutions for designing, developing and researching next generation exertion interfaces [Mueller et. al, 2002, 2008].

## 2. BACKGROUND

As discussed in [Sheridan, 2010], in the 1940s, Iona and Peter Mason Opie began collecting, archiving and classifying hundreds of nursery rhymes and songs from written sources – including the known facts about origin and variants. Their seminal publications, such as *The Oxford Dictionary of Nursery Rhymes* [Opie & Opie, 1951], have influenced throngs of folklorists. In the 1950s they corresponded with teachers to explore school-aged children's lore and produced *The Lore and Language of Schoolchildren* [Opie & Opie, 1959]. In later field work and publications, such as [Opie & Opie, 1985], the Opies refuted the idea that mass media negatively affected or extinguished traditional games and their vibrant testimony and audio archives stands as lasting proof that children's traditional games thrive in streets and playgrounds.

Our project seeks to identify how traditional playground games can be adapted using new technologies, namely the Nintendo Wii. The intention is to capture and visualise real-time 6DOF movement of clapping games as well as to allow children to archive their own clapping games.

In [Sheridan, 2010] we identified preliminary problems and potential solutions for adapting Wiimote games, namely the Physics of clapping - that clapping games concern tracking of extremely fast hand movement in 6DOF (Figure 1).

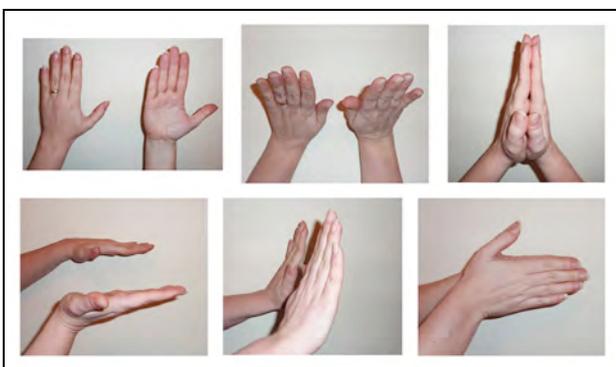


Figure 1. Three-way clapping game with hands in 'flipped' and mirrored positions.

In this paper we highlight the iterative technical development, and focus on the adaptation of a game controller and development of the interface as well as identify the problems and solutions concerning 6DOF tracking. Our goal is to identify solutions for developing a low-cost, low-fi and highly mobile prototype system for which other researchers could use for capturing extremely fast movement of two hands in 6DOF using the Nintendo Wii as a baseline platform.

Important to note, is that our challenge was in plotting *absolute* rather than *relative* positioning. In other words, our solution required that we map the exact location of hands in the real world with hands in the virtual world in space and time (absolute position) rather than an indirect mapping (relative positioning). In the next sections, we illustrate these constraints and describe techniques for overcoming them.

## 3. UNDERSTANDING 6DOF

Six degree of freedom motion tracking is a multipart problem usually solved through using complex 3D digital, optical and/or body-worn technologies. For example, the Viacon system [Viacon] uses bespoke software, multiple high specification cameras and optical hardware as well as body-worn markers to track and visualise human motion. While many of these complex systems offer robust and accurate 6DOF metrics, they can be costly, require extensive set up and/or require installation in a permanent space. On the other hand, the EyesWeb system [EyesWeb] offers an open source solution using one or more videocameras and/or sensors for visualising expressive gesture. This system is good for identifying low-level motion cues, for example, how much time a given position in the space has been occupied rather than tracking full 6DOF human motion.

### 3.1 Tracking human motion

Tracking motion in 6DOF requires an understanding of how humans move in 3D space and how this movement is translated in hardware and software systems. We can represent human motion on occurring in 3D on three axes (Figure 2).

For example, a person walking forward and backward would produce data on the Z axis (represented by the green arrow). Walking forward would produce data in the +Z axis and walking backward would result in movement on the -Z axis. Similarly, walking left or right would produce data in the X axis or up and down would produce data in the Y axis. This is referred to as *linear movement*.

In order to track movement in 6DOF, we also need to track movement moving *around* each axis. We call this *rotation*.

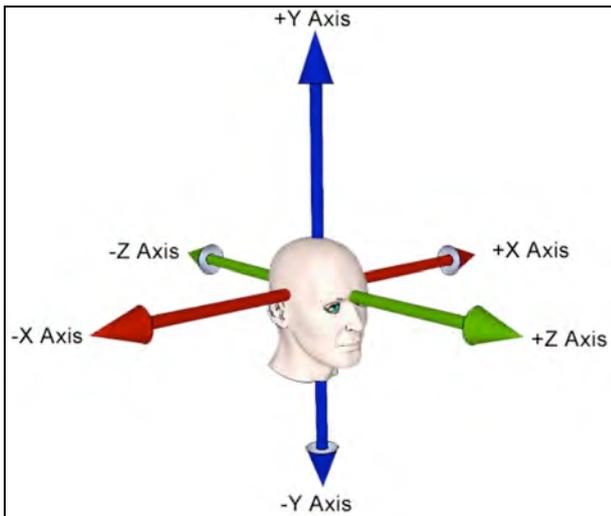


Figure 2. Visual representation of X, Y, Z axis.

For example, nodding the head forward and backward produces data about the head rotating around the X axis. This movement is known as *pitch*. Rotations around the Z axis are called *roll* and rotations around the Y axis are called *yaw*.

### 3.2 Tracking Wii linear movement and rotation

The Wiimote uses infrared technology (IR) to track linear movements in 2D (X and Y axis) and an internal accelerometer to track rotation (Figure 3).

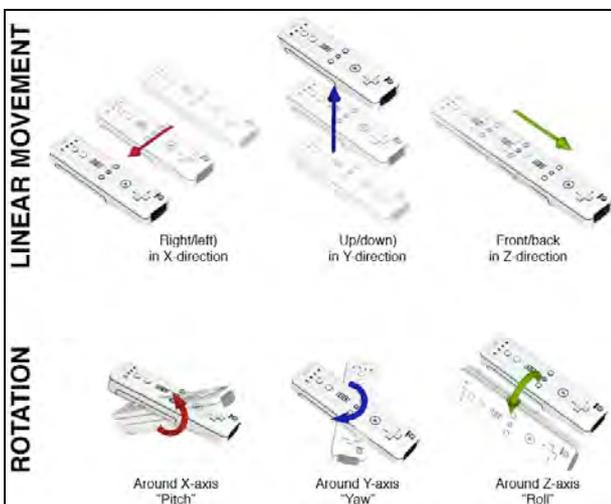


Figure 3. Wiimote linear movement and rotation.

However, a number of physical and technical constraints limit the potential for using the Wiimote alone to track full 6DOF motion. First we describe a method and code for capturing Wiimote data, and then describe the physical and technical constraints.

OSCulator [OSCulator] is an open source application that can be used to capture data from a

Wiimote controller. OSCulator uses the OSC protocol [Open Sound Control] to pair the Wiimote with the OSCulator interface via Bluetooth. The OSCulator GUI interface allows users to: connect multiple controllers at the same time; specify which axis, button or rotation they would like to capture data from; smooth data streams to prevent jittering; assign events and values to data streams; and, to visualise multiple raw data streams on a 2D graph.

For our purposes, it was necessary to then write a small application to visualise the raw data in 3D. In other words, we needed a way of plotting the linear motion and rotation data (6DOF) streaming from our Wiimotes onto a 3D environment. To do so, we wrote a small application using Processing [Processing]. Processing is an open source programming language and environment for creating images, animations and interactions. Using Processing, we were able to write a small application to import Wiimote data from OSCulator into Processing. Once this step was completed, we then modelled two 3D hands using the open source 3D modelling tool Google SketchUp [Google] and extended our Processing application to assign each hand to each Wiimote. In other words, moving the left Wiimote in physical space caused the left hand to move in virtual space and moving the right Wiimote caused the virtual right hand to move. We rendered this movement in a 3D environment we wrote in Processing using OpenGL specification [OpenGL]. However, we quickly recognized a translation problem, which we explain in the next section.

The Wiimote uses an internal accelerometer to track rotation. Because of the placement and calibration of the accelerometer, the correct initial position for capturing real-world Wiimote data is position the Wiimote as illustrated in (Figure 4, left). Since our intention was to capture the action of clapping – we needed to either place the Wiimote on the top of a glove or participants would hold the Wiimote in their hand. This meant that we needed to *translate* our data streams in our Processing application in order to capture the correct linear movement and rotation data from our Wiimote (Figure 4, right).

Once our translation was working sufficiently, we detected three additional issues to overcome: 1) tracking depth; 2) rotation beyond 180 degrees; and (Figure 5), 3) two intersecting paths (Figure 6).

The Wiimote uses IR (infrared) sensing to track linear movement in the X and Y axis – as long as the Wiimote is pointing towards the screen, linear movements can be easily detected in the X and Y axis. However, once translating and repositioning our Wiimote, as illustrated in Figure 5, the Wiimote

is unable to detect depth - forward and backward linear movements (i.e. movement on the Z axis).

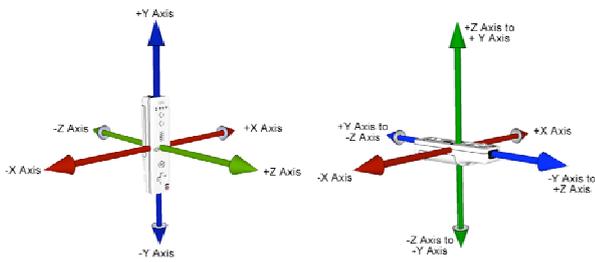


Figure 4. Correct orientation of Wiimote in relation to accelerometer placement and calibration without translation (left) and with translation (right).

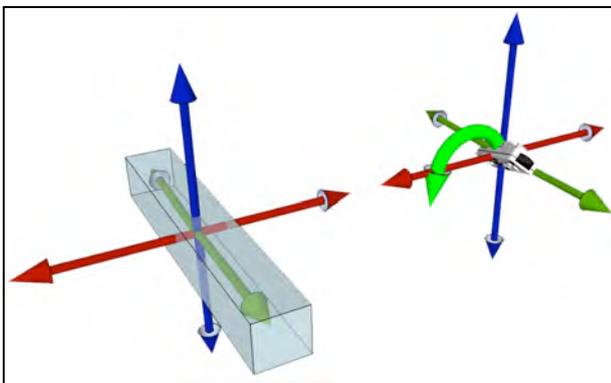


Figure 5. Difficulties in using a Wiimote for visualising 6DOF motion control: depth (left) or detection of rotation over 180 degrees (right).

In the commercial system, the Wiimote overcomes this obstacle by, for example, using a combination of IR tracking, hitting a physical button on the underside of the Wiimote, and using the accelerometer data to simulate depth. For example, in WiiBowling, when players swing the Wiimote forward they are required to hit the 'B' button to 'release' the ball forward. In WiiSports the virtual Wiimote is often constrained to a limited space so that the virtual bat or paddle only ever moves left to right.

Since one of our challenges was to mount the Wiimote on a glove on top of the hand and that we needed to capture 6DOF, we had to consider that the Wiimote would be positioned in a ways that the buttons would not be accessible and that the hands would be moving both forward *and* backward in unconstrained space.

In addition, due to its configuration, the Wiimote is unable to detect when it has flipped more than 180 degrees (Figure 5). Since many playground games begin with at least one hand flipped completely over, as in a 3-way clapping game (Figure 1), we

needed to find a solution that would overcome this limitation.

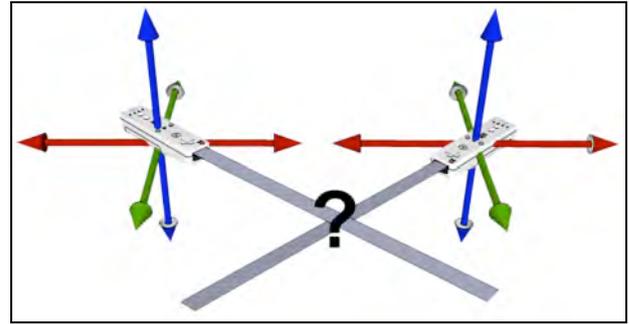


Figure 6. Wii system is unable to distinguish IR from multiple Wii.

Finally, as discussed above, playground clapping games involve two hands, so our solution had to include two-handed input and that any solution needed to take into account that in clapping games: hands have a point of impact; they cross over; and, they mirror each other at a very rapid pace. Since a Wiimote uses IR to detect movement on the X and Y axis, the Wiimote system is unable to detect when the IR from two Wiimotes intersect as illustrated in (Figure 6).

At this stage, it became clear that the Wiimote alone (without using buttons as an aid) could not physically overcome these known issues. Our solution then was to attempt to develop a vision tracking solution in combination with the Wiimote. We describe this development in the next section.

#### 4. VISION TRACKING METHOD

Our investigation involved the iterative development of several different prototypes, each of which revealed important and fundamental issues concerning how to design technologies for 6DOF motion tracking. In this section, we describe the various prototypes and discuss why we made particular design decisions. Using information gathered from our first user tests [Anonymous], we decided to test a vision tracking method to try and overcome the problem of tracking the depth and intersection points.

##### 4.1 Blob tracking with a camera

For our first prototype, we identified that we could easily track linear hand movement depth using an external camera (Logitech Quickcam Pro) placed on the floor facing upwards (Figure 7).

A camera with mid-level frames per second (15fps) is sufficient as most computers are unable to handle HD camera resolution (1920x1080, 30fps) particularly when developing a Processing

application. A wide-angle lens is useful for detecting a larger interaction area. In addition, vision-tracking applications require that room lighting is controllable (i.e. low-level lighting or no direct light/sunlight in a camera lens).

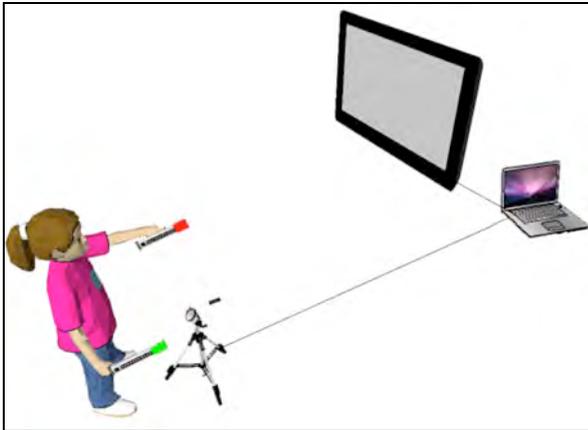


Figure 7. Tracking linear depth (Z axis) movement with camera on the floor facing upward.

For our vision tracking application, we used the open source OpenCV library originally developed by Intel [OpenCV]. OpenCV contains a blob detection library [OpenCV blob detection] that allows users to detect points or regions in an image that are brighter or darker than the surroundings. Our intention was to use blob detection to track the size of the blob and use this measure to estimate movement on the Z axis. Additionally, we determined that one method that could be used to distinguish the movement of different hands was to track two different coloured blobs (where each hand was a different coloured blob). Since OpenCV does not have a colour-tracking library, we wrote a bespoke extension for the OpenCV library which would allow us to track multiple and different colours in real time.

For our first test, we used coloured paper wristbands (Figure 8, top left) of various colours, materials and sizes, however a small floor lamp was needed to make the wristbands bright enough for the camera to detect the different colours.

In addition, since the camera picked up colour from its surroundings (e.g. skin colour), long black gloves were needed to occlude tracking colours from the arms (Figure 8, top left) as well as a slider to allow us to adjust the brightness/saturation so that we could easily adapt our application according to the variable lighting conditions in the environment. We then built and tested several different prototype wristbands using coloured LEDs (Figure 8, top right, bottom).



Figure 8. Gloves and paper wristbands (Left, top) and various LED wristbands.

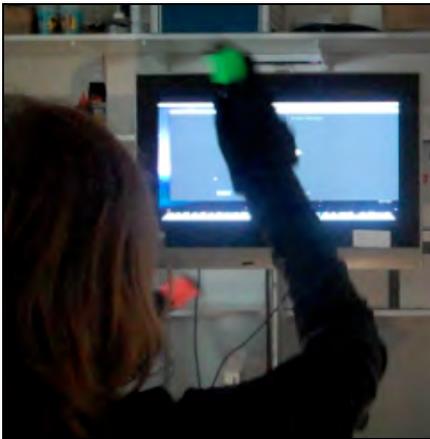
#### 4.2 Issues with using LEDs

A number of factors affect the kinds of LEDs that can be used: brightness, colour, size, shape, and number. In terms of colour, ultrabright LEDs (Figure 8, bottom left) do not work well, even in complete darkness because they are too bright (so the camera only sees the colour white) and likewise, many blue LEDs are seen as white. We eventually identified that green and red are the best colours to track – they are far enough apart on the colour spectrum to reduce the probability that the camera would identify the wrong colour. Brightness and colour also affect the size of the LED needed. For example, because the colour red is brighter than green, a red LED will appear bigger even when it is the same size as the green LED (so, the camera will assume that the red LED is closer to the camera than the green one, and this will then cause errors in tracking). Flathead LEDs (Figure 8, bottom right) work best as they diffuse light in a more even manner than roundhead LEDs (Figure 8, top right) - the tip of a round LED is much brighter than the sides of the LED (creating a white spot) whereas the top of a flathead LED distributes light evenly and across a larger area. Different shaped LEDs have a different viewing angle so it is necessary to consider how the LEDs will be angled and positioned. Also, it is necessary to determine the right spacing between LEDs on a board (Figure 8, top right) so that the camera tracks the multiple LEDs as one whole object, rather than multiple objects. A final point to consider is that LEDs consume battery power, and a switch (Figure 8, bottom right) is useful for turning them on or off.

#### 4.3 Informal usability studies

After testing the LEDs with our Processing application, we conducted several informal studies with project team members as well as with several

children between the ages of 8-11, both boys and girls from two schools – one in Sheffield and one in London (Figure 9). For the studies, the space was set up as described in (Figure 7).



*Figure 9. Children testing the vision-tracking application.*

Participants were given a brief explanation and demonstration of how the system worked. One investigator remained in the room with the participants while they played with the interface to help the participants put on the gloves/wristbands, to take observational notes, and to take video documentation.

For each session, one participant would play at a time with the interface while another person watched and waited for their turn. While the investigator asked specific questions, the participants were free to ask any questions they wanted to at any time during the test. Participants who were not in the Wiimote testing room during the test, were engaged in two other activities about playground games: interviews with two ethnographers; and, drawing activities with ethnographers. All activities were videotaped for later analysis.

The investigator asked two different sets of questions: movement questions and interface questions. The movement questions were designed to have each participant perform repeatable movements to examine the ease of use and robustness of the system. For example, try moving one hand up and one hand down; try moving your hands together; try clapping; try crossing your hands over. The interface questions centred around the look and feel of the user interface, for example: How does it feel? Is it easy to use? What would you change?

#### **4.4 Summary of findings from study**

During the tests, a number of new issues emerged: scaling, trajectory, range and distance. Scaling, or the size of the virtual hands, changed depending

on lighting as well as camera placement. In accordance with the laws of Physics, brighter objects appear closer to the camera and less bright objects appear further away. So in our application if environmental lighting suddenly became brighter, then the LEDs appeared less bright, and therefore the virtual hands would appear further away even if they had not moved at all. This confused participants. We determined that we needed to create an offset variable in our Processing application so that we could change where the hands appeared on the screen depending on variable lighting conditions in the environment.

The nature of rendering and moving virtual objects in 3D environments combined with the camera barrel distortion, meant that when participants moved their hands on the Z axis in real space, the virtual hand appeared to move around an invisible 'spherical space', or to slightly pitch upwards – there was a noticeable curved trajectory in movement. This issue also confused participants – they couldn't understand why their virtual hand would pitch upwards when they perceived their movements as flat and in a straight line.

Participants had some difficulty in keeping their movements within the intended 'zone of interaction'. If they moved their hands beyond the range of the camera lens, then their virtual hands no longer appeared on the screen. In addition, the optimal distance for camera placement and the hands depended on the lighting conditions in the room. When uplighting was used, the optimal standing distance was approximately 3-5 feet from the camera lens with the camera placed at knee length. Confining movement to a rather small space (arm's length) was slightly cumbersome and awkward for many of children; we observed that when children were given the task of "moving around in any manner they wished" they tended to want to move their bodies around in unconfined space even when they were explicitly told that the system wouldn't work if they moved outside a particular 'zone of interaction'. Children were less interested in fine-tuning their movements or adapting their 'natural' movements to suit the tasks than they were at making the virtual hand move around on the screen.

An interesting observation can be made from the participants' mental mapping of 6DOF in this situation. This ambiguity of movement often caused children to be physically creative [Anonymous] and enticed performative interaction [Sheridan & Bryan-Kinns, 2008]. Adults, on the other hand, tended to remain within the confined space restrictions. It was the adults, rather than the children that expressed a desire to have even more fine-grained control over hand movements. Adults were often frustrated that the hands showed any movement other than what

they would expect to perceive as an exact copy of movement in the real world. Conversely, the 6DOF semi-indirect mapping caused children to make up their own games, to talk to each other and to use their imagination.

In addition, the children didn't mind wearing gloves in the test, however they did require the investigator to help them put them on. In order to make the system more robust, we considered that we would need to refine our solution so that wearing gloves was not a necessary part of the set up.

Our informal usability study confirmed that LEDs were useful for tracking the linear motion of multiple hands, of different colours, at high speeds. However, the testing also revealed problems with our set up and striking a balance between physical creativity and fine-tuning motor control. We describe how we resolved these problems in the next section.

## 5. COMBINING VISION TRACKING + WIIMOTE

Our next step was to combine linear motion tracking using our LEDs motion tracking application with Wiimote rotation to create a more robust 6DOF tracking method and to increase ease of use by not using gloves or an external camera. From our earlier tests, we determined that we should evenly diffuse light across a large surface area. Since we were now using the internal camera (i.e. on a MacBook Pro laptop), the flathead LEDs (Figure 8, top right) were no longer large enough or bright enough for our purposes. As such, we needed to build two large coloured LED objects to track that could be placed on the end of the Wiimote.



Figure 10. Ping-Pong LED balls for colour tracking.

Our solution was to wrap three large round-headed LEDs in semi-transparent paper (i.e. tissue paper or toilet paper) to evenly diffuse the light. We then taped the LEDs inside the bottom of a Ping-Pong ball (Figure 10) and then taped a 5V battery to the end of the LED legs using black tape; using black tape is important since it absorbs light rather than reflects it thus reducing the number of potential

false recognitions from colours in the surrounding environment.

The Ping-Pong balls were then attached to the end of the Wiimotes (which were also wrapped in black tape to avoid any reflection). The result was two large glowing and coloured orbs whose colour could be easily distinguished and tracked by a standard webcam as depicted in (Figure 11). The solution resembles a DIY version of the Sony Move controller [Sony].

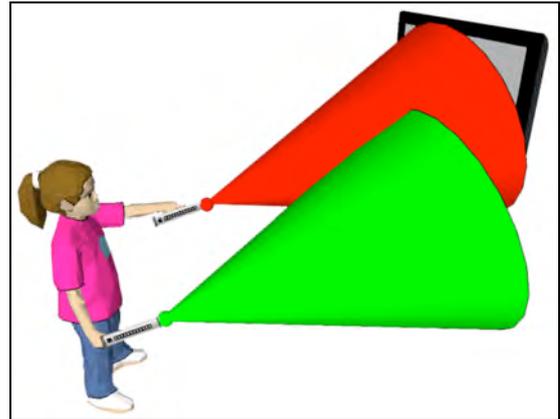


Figure 11. Ping-Pong LED balls + Wiimotes for multi-hand 6DOF colour tracking.

Since the camera placement was now in front of the user, rather than on the floor, it was then necessary for us to again translate linear movement and rotation in Processing to suit our new set up. We extended our application so that both the *size and the shape* of the green or red ball in the X or Y axis represented movement on the Z axis. For example, if the camera saw a small circle, then the ball must be far away (-Z axis) and if the ball becomes bigger than it must be moving forward (+Z axis). However, in order to adjust the camera to the correct lighting conditions of the environment and the LEDs, a calibration process was required. To make this process easier, we then extended our application to include an automated calibration process. Participants would simply hold the modified Wiimote with glowing orb in front of the camera and move it slowly up and down and back and forth. The application was then able to subtract any potential colours that were outside of the coloured circle, thus making the system far more robust.

Additionally, we changed the look of the virtual hands so that they were more realistic. A floor was added to the 3D environment to help participants orientate the virtual hands in relation to the real world. In the next section, we describe the iterative development of the software architecture and interface as well as the data visualisation.

## 6. INTERFACE

Our previous system [Sheridan, 2010] described our first prototype application for capturing, playing, re-playing and visualising data. Here we detail the further incremental development of our simple game architecture.

### 6.1 GUI

We integrated our new 6DOF tracking application with a graphic user interface (GUI), which included dynamic buttons that bounced when the Wiimote hovered over them. Since the Wiimotes were now placed in the hand, participants could use the buttons to select various options presented on the screen. Graphic elements, such as images and text, were created using GIMP [GIMP]. The new interface presents participants with three options: 1) Recording a new game; 2) Playing against the computer; or, 3) Viewing player's saved games (Figure 12).

Like the previously developed system, recording a new game allowed participants to record their hand movements on the screen and then replay on the screen as many times as they wished. Playing a game against the computer allowed participants to play against a pre-recorded set of moves with an accompanying song (e.g. "I Went To A Chinese Restaurant", see [Sheridan, 2010]). At any point during the game, participants can save their game, and then view their saved games. In addition, a home and help button was added to ease navigation errors identified in previous studies.



Figure 12. GUI game interface.

Once an option was selected, participants were prompted with dynamic text. For example, when beginning to record a game the interface would count down 'Get Ready' '3' '2' '1' 'Go' to allow participant to prepare themselves for the beginning of the game (Figure 12).

### 6.2 Visualisation of data

In addition, the interface provides a research visualisation tool in our Processing application for

allowing researchers to view any of the saved 6DOF movements. So for example, movements from a clapping game can be visualised as shown in Figure 13. In addition, visualisations can be rotated in real time in 3D space so that it can be viewed from any angle (Figure 13).

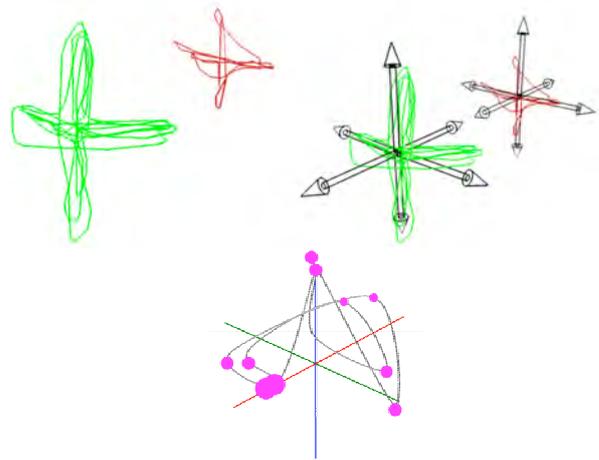


Figure 13. Visualisation of 6DOF raw data where green represents the right hand and red the left hand (left) and plotted on 3 axes for better viewing (top, right). Visualisations can be rotated in 3D to view from any axis or smoothed (bottom) - pink dots represent "points of impact" or "change in direction."

Once our system was robust enough for testing, we conducted a pilot usability study with several adults. The outline of the study was identical to that described in Section 4.

### 6.3 Informal usability studies

We conducted several informal studies on 3 different days, at different times of the day, with adults as well as with several children between the ages of 6-12, both boys and girls from the London area, mostly family and friends. Again, the space was set up as described in (Figure 7) and participants were given a brief explanation and demonstration of how the system worked. One investigator remained in the room with the participants while they played with the interface and to take observational notes.

Also, we again employed a turn taking versus spectator methodology for each session as we noted that children seemed more comfortable and less shy in pairs than on their own. For each session, one participant would play at a time with the interface while another person watched and waited for their turn. Since the tests were conducted on different days, we did not arrange to have additional activities happening outside the room. Some of the tests were not videotaped at the

request of the parents. Our investigator questions remained the same: movement questions and interface questions.

## 7. KINESTHETIC LITERACY

Applying “kinesthetic literacy” [Sheridan & Mueller, 2009], or learning to move and moving to learn, as a guideline for designing for physical creativity allows participants to know the limitations of both self and other, giving them an opportunity to adapt play according to these limitations.

In combining colour, size of the orb with the rotation of the Wiimote, the system allowed participants to move their bodies in a more ‘natural’ and ‘fluid’ way. For example, our system was able to respond to very quick hand movement in an (perceived) immediate and direct way (“Look! When I move my hands, those hands move too!”). In every test, the participants began to try and work out how the system was working (“How does it do that?” or “Does it have a motion tracking sensor in it?”). In every case, the investigator would push the participants to figure out the answer for themselves (“How do you think it works? What if you try moving like X, what happens? Can you figure out why?”). This caused participants to challenge each other (“How fast can you do it? Faster, faster!” or “I’m going to do it better than you.”)

The calibration process worked extremely well in reducing the number of false recognitions. However, since the balls were slightly bigger than the width of the end of the Wiimotes. On occasion, participants would knock the balls against each other causing them to nearly fall off once or twice. In between tests and once during a test, we had to use tape to make our system a bit more robust. Despite this, there was a huge improvement in the system responding to very quick changes in direction – or the Physics of clapping, and “punctuation” [Sheridan, 2010].

The addition of a virtual floor in the 3D environment aided participants in orientating and positioning their hands in 6DOF in the real as well as virtual world. The elimination of the gloves and the external camera meant that participants were able to simply pick up the Wiimotes and immediately start using the system. A number of improvements could be made to the system which we outline in the discussion section.

## 8. DISCUSSION

We have developed a prototype open source 6DOF tool and interface for multi-dimensional and multi-hand tracking as well as visualisation.

At the time of project completion, open source code for developing applications for the Microsoft Kinect has been released through the open source community [KinectHacks, OpenKinect]. Like our research, the Kinect offers a vision-tracking solution. Whether the Kinect is robust enough to track, for example, the problems we identified in our research (i.e. depth, rotation over 180 degrees and intersecting data from multiple hands) has yet to be seen (at the time of writing). Our intention for our next stage of development is to combine the methods, tools and code described in this paper with Kinect technology [Microsoft]. We suggest that this combination could provide a robust 6DOF motion tracking and visualisation tool for innovative and next generation exertion interfaces.

In designing exertion interfaces for 6DOF a number of remaining questions emerge: how do we strike a balance between participants perceptions of 6DOF movements in the real-world and mental mapping with movements in the virtual world? How important is *movement realism* in our games? What can researchers learn from visualising 6DOF movements? The recent publication of the Exertion Framework [Mueller et. al, 2011] may provide answers to some of these questions.

A number of different skills were required for designing for exertion games: software development, usability studies, user experience modelling, 3D modelling, among others. Will these new motion-tracking systems require new kinds of skill sets from individual designers [Paradiso et. al, 2008]?

We look forward to seeing how the open source community further develops, extends and re-writes the applications and tools generated from the research presented in this paper.

## ACKNOWLEDGEMENTS

Thank you to BigDog Interactive for ongoing support and for providing the prototyping tools. The open source code, documentation and videos are available at SourceForge and at jennifer [dot] Sheridan [dot] com. The 3D SketchUp models are available on the Google 3D Warehouse. Many, many thanks to the open source community.

## REFERENCES

Benovoy, M.; Zadel, M.; Absar, R.; Wozniowski, M.; and Cooperstock, J. R. Towards immersive multimodal gameplay. In GAMEON-NA, Montreal, QC, Canada, August, 2008.

Bradshaw, D and Ng, K. Tracking Conductors Hand Movements Using Multiple Wiimotes. In

Proceedings of the International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution. 19-19 Nov. 2008, Washington D.C., pp. 93-99.

Code available at SourceForge.

England, D., Randles, M., Fergus, P. and Taleb-Bendiab, A. Towards an Advanced Framework for Whole Body Interaction. In Proceedings of the 3<sup>rd</sup> International Conference on Virtual and Mixed Reality/HCI International 2009, Springer-Verlag, Berlin.

EyesWeb. [musart.dist.unige.it/EywMain.html](http://musart.dist.unige.it/EywMain.html) [Last checked: January 17, 2011].

GIMP. [www.gimp.org/](http://www.gimp.org/) [Last checked: January 17, 2011].

Google SketchUp. [sketchup.google.com/](http://sketchup.google.com/) [Last checked: January 17, 2011].

Google 3D Warehouse  
<http://sketchup.google.com/3dwarehouse/> [Last checked: January 17, 2011].

Kinect Hacks. [kinecthacks.net/](http://kinecthacks.net/) [Last checked: January 17, 2011].

Lee, J.C. Hacking the Nintendo Wii Remote. *Pervasive Computing*, 7(3), July-Sept. 2008, pp. 39-45.

Microsoft Kinect. [www.xbox.com/kinect](http://www.xbox.com/kinect) [Last checked: January 28, 2011].

Mueller, F., Agamanolis, S. and Picard, R. 2002. Exertion interfaces for sports over a distance. In *Proc. of UIST'02* (Paris, France, October 27-30, 2002). ACM, New York, NY.

Mueller, F., Gibbs, M. R. and Vetere, F. Taxonomy of Exertion Games. In *Proc. of OZCHI'08* (Queensland, Australia 2008), 263-266.

Mueller, F., Edge, D., Vetere, F., Gibbs, M. R., Agamanolis, S., Bongers, B., & Sheridan, J. G. (2011). Designing Sports: A Framework for Exertion Games. In CHI '11: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vancouver, Canada.

Nintendo Wii Remote. [www.nintendo.com/wii](http://www.nintendo.com/wii) [Last checked: January 17, 2011].

OSculator. [www.osculator.net](http://www.osculator.net) [Last checked: January 17, 2011].

OpenCV for Processing.  
[ubaa.net/shared/processing/opencv/](http://ubaa.net/shared/processing/opencv/) [Last checked: January 17, 2011].  
OpenCV blob detection.  
[ubaa.net/shared/processing/opencv/opencv\\_blobs.html](http://ubaa.net/shared/processing/opencv/opencv_blobs.html) [Last checked: January 17, 2011].

OpenGL for Processing.  
[processing.org/reference/libraries/opengl/](http://processing.org/reference/libraries/opengl/) [Last checked: January 17, 2011].

OpenKinect.org. [openkinect.org/wiki/Main\\_Page](http://openkinect.org/wiki/Main_Page) [Last checked: January 17, 2011].

Opie, I. and Opie, P. 1951. *The Oxford Dictionary of Nursery Rhymes*. Oxford University Press, Oxford, UK.

Opie, I. and Opie, P. 1959. *The Lore and Language of School-children*. Clarendon Press, Oxford, UK.

Opie, I. and Opie, P. 1985. *The Singing Game*. Oxford University Press, Oxford, UK.

Paradiso, J.A., Heidemann, J., and Zimmerman, T.G. Hacking is Pervasive. *Pervasive Computing*, 7(3), July-Sept. 2008, pp. 13-15.

Processing. [processing.org](http://processing.org) [Last checked: January 17, 2011].

Sheridan, J.G. and Bryan-Kinns, N. 2008. Designing for Performative Tangible Interaction. *Int. Jour. of Arts and Technology*. 1(3/4), 288-308.

Sheridan, J.G. and Mueller, F. 2010. Fostering Kinesthetic Literacy through Exertion in Whole Body Interaction. In *Proc. of the Workshop on Whole Body Interaction, CHI'10*, (Atlanta, USA, April 10-11, 2010).

Sheridan, J.G. When Clapping Data Speaks to Wii: Physical Creativity and Performative Interaction in Playground Games and Songs. In Proceedings of the 24th BCS Conference on Human Computer Interaction. 6-10 September 2010, Dundee, Scotland.

Sony Playstation Move.  
[uk.playstation.com/psmove/](http://uk.playstation.com/psmove/) [Last checked: January 17, 2011].

Vicon. [www.vicon.com/](http://www.vicon.com/) [Last checked: January 17, 2011].