Capturing and Visualising Playground Games and Performance: A Wii and Kinect based motion capture system

Grethe Mitchell
University of East London
4-6 University Way
London E16 2RD
United Kingdom
g.r.mitchell@uel.ac.uk

Andy Clarke
78 West Kensington Court
Edith Villas
London W14 9AB
United Kingdom
andy@kinonet.com

In this paper we present the design and development of an interactive application which uses open source software (Processing, libfreenect, DarwiinRemote, OSC) and hacked games hardware (both the Microsoft Kinect sensor and Nintendo Wiimotes) to create a low-cost markerless motion tracking system that allows the recording, playback, visualisation and analysis of movement in 3D, using children's clapping games as an example. This fully-functional proof of concept provides researchers in the arts and humanities with a new and innovative way of visualising, analysing and archiving gesture and movement such as clapping games, and opens up possibilities for other applications in movement, music and the performing arts.

Playground games. Clapping games. Kinect. Wii. Motion capture. Motion tracking. Visualisation. Game controllers. Computer games. Performance.

1. INTRODUCTION

In this paper we introduce and describe a low-cost and robust markerless motion tracking application for capturing, visualising and analysing movement, constructed with easily available computer game hardware and open source software. Developed to enable children and researchers to record and playback game and play activities (with sound and text, where required), and as a research and visualisation system, the Game Catcher can be used with other activities and has applications in a wide range of disciplines within the arts and humanities, including in performance and music.

2. OVERVIEW OF PROJECT

The Game Catcher is a motion tracking research tool and computer game, developed as part of the "Playground Games and Songs in the Age of New Media" project (Burn et al., 2011). This two-year project was funded by the AHRC, as part of the Beyond Text Programme.

The "Playground Games and Songs" project as a whole involved four institutions: the Universities of East London, London, and Sheffield, and the British Library. The development of the Game Catcher

was supervised by Grethe Mitchell of the University of East London, and the final version of the application was developed by Andy Clarke.

In producing the Game Catcher, we had two main aims. The first, and most relevant to this conference, was to provide a way for the physical movements of clapping games to be recorded and analysed by researchers, thereby providing a proof of concept for the use of movement capture technology in general in the humanities. The second aim was to "port" a real-life playground game to a computer game so as to better understand the differences between them, as well as points of similarity. We describe each these aims in more detail in the next section.

3. GAME CATCHER AIMS AND CONTEXT

3.1 Archiving and Analysis

There are many disciplines and areas of study within the arts and humanities which would benefit from recording, archiving and analysis of movement. These include areas such as dance, drama and music teaching, learning and performance, childhood development and play/games.

Nonetheless, the techniques for recording and documenting movement are patchy, with bodies of knowledge siloed within certain fields and little known (or little used) outside of that particular field. For instance, formal movement notation systems such as Labanotation are used within dance, but not outside of it, even though it could have application elsewhere. Commercial motion capture systems are likewise used in the entertainment industry and in high-end medical or sports research and development, but are not generally used in the arts and humanities.

In general, any system for recording movement can be assessed by a number of criteria. Firstly, there is the issue of ease of use – which applies to both the researcher and the subject. For the researcher, there are issues of whether the system requires an excessive level of knowledge, training, setup or effort on their part. For the subject, there is the issue of whether the system encumbers or restricts their movements or otherwise inconveniences them (for instance, by requiring them to pause or repeat their movements while they are being recorded).

Then there are issues such as accuracy (how precisely movements are recorded), fidelity (how closely the recording portrays events), resolution (the level of detail recorded) and completeness (whether significant moves are omitted). Resolution can furthermore be split into spatial resolution (how precisely details can be observed) and temporal resolution (how frequently measurements are taken).

Although these properties are related, it is important to note that a good performance in one does not necessarily mean similar performance in another. A series of still photographs, for example, will have high accuracy, but a low temporal resolution and a low level of completeness (as there are gaps in time between the photographs). Conversely, an animation may have high resolution and completeness, but low accuracy (unless it was produced by rotoscoping).

The properties also are not fixed, but also depend, in part, on how the system is used. With regard to its recording of movement, a video will have a high spatial resolution if it is taken from close up, and a low spatial resolution if it is taken from far away (even though the accuracy and fidelity remain constant) – in other words, it is dependent on viewpoint. This resolution is not necessarily constant as the camera may, for example, move closer to the action during the course of the sequence. Similarly, a series of photographs may vary its temporal resolution if instead of taking photographs at regular intervals, there are more

photographs taken at a time of faster or more significant movement.

Flexibility is also an issue, both with regard to the type of movement that can be recorded and to the uses to which data can be put afterwards. Confidentiality and anonymity can also be of concern, particularly when dealing with children, and even if one takes care to leave the child's face obscured, it is easy to inadvertently leave in other details which might identify the school or location.

We felt that a low-cost markerless motion tracking system would be able to combine the ease of use of video, whilst overcoming its potential shortcomings with regard to spatial resolution and flexibility. Video provides a level of spatial resolution which is chosen at the time of shooting (through the choice of viewpoint) and which cannot subsequently be changed. This viewpoint may, therefore, either be too close and leave essential details off-screen or too far away and leave them indistinct. Motion capture, on the other hand, allows the movement to be viewed from any angle or distance, not just the one that it was recorded from. By recording raw movement data in a computer-readable format, there is the potential for it to be subsequently transformed into any other notation system (e.g., Labanotation); it can also be used to generate animations which faithfully portray the subject's movements, but leave them unidentifiable. Furthermore, if the motion capture system was markerless, it would not encumber the user, nor require any lengthy setup on the part of the researcher.

The Game Catcher was therefore developed with these ideas and principles in mind and intended to act as a fully functional proof of concept of such a system. Another design principle that we had in mind was that the system should be low cost and robust, and this led to our decision to use modified game hardware to create the Game Catcher. This is discussed in greater depth later in the paper.

3.2 Game Prototype

The second main aim of the Game Catcher project was to “port” a real-life playground game to a computer game as the process of doing this would force us to think more formally about what a playground game consists of and about the relationship between physical and virtual games. (Mitchell, 2010a & 2010b). In the case of the clapping game, this raises questions such what its vocabulary of moves are. More generally, one can think about what components can be removed, reduced, enhanced, and added, as well as whether any can be substituted or combined. In the context of the project, it was also conceived as a form of

“cultural intervention” positioned to investigate the differences between physical and virtual forms of play, and to investigate the possibilities of producing a modifiable and open-ended game application.

But it was not just an intellectual exercise – we also wanted to make a game which was enjoyable, as this would generate additional positive synergies. We therefore regarded it as important to combine these two functions – game and archiving tool – in one application as this allowed us to explore these synergies and exploit them to the full. We envisioned that it would be possible to create a virtuous circle with the Game Catcher, as shown in figure 1.

Figure 1: Synergies between play and research in the Game Catcher application

As the children played the game, it would record their movements. This would form raw data which could then be analysed by the researchers. The data would also form part of the game itself (or series of games), adding to the library of pre-recorded games and thereby adding depth and variety to the games and making them more appealing to children.

These effects would be particularly strong when, over time, the Game Catcher had been taken to a greater number of locations. Clapping games (as a genre of game) are very widespread, but individual variants of them can be geographically and temporally isolated. Pupils at one school may, for example, not be aware of a clapping game played at another school in the same city unless there is a mixing of their pupils outside of school. Likewise, children may not know the version of a clapping game or song played at their own school a few years previously, as the rate of evolution and mutation of clapping rhymes may be rapid. Clapping games were chosen for the first version of the Game Catcher as they offer challenges, but also have constraints which make these challenges manageable within the timescale and budget of the

research project. With regard to the challenges, clapping games feature fast and unpredictable hand movements, with a high potential for occlusion or misrecognition; they also require a tracking system which doesn't impede the player excessively. On the positive side, clapping games take place within a limited playing area, with a player standing still and just moving their hands. Clapping games also have some conventions about how the hands move, with certain hand positions, orientations and movements being common and others not used.

4. THE PROCESS OF ADAPTATION

As we began the process of adapting real-life clapping games to a computer game, we began to consider the most appropriate terminology for it. One can, for example, think of it as “porting” – in the sense that a piece of software is ported from one operating system to another. This implies that having a similar appearance/behaviour is most important, even though what is happening behind the scenes may be different. “Adaptation”, on the other hand, implies something different – it suggests that what is important is the sense or the feeling, rather than the look, and that some degree of flexibility allowed in order to achieve this. “Translation” tends to imply more of a literal process, while “inspired by” is at the other extreme, suggesting that there is a relatively weak connection between elements in the source material and those in the final product.

The difficulty in finding the ideal term stems in part from the fact that although these terms each come from different areas, they all apply, for the most part, to the process of transferring an object from one field to another, not something as ephemeral as a game.

Perhaps the most appropriate term for the process is that developed by Kress (2009 pp47) where he describes as *transduction* the process whereby something which has been configured or shaped in one set of modes (e.g. playground games) is then *reconfigured* and *reshaped* according to the affordances of a different mode (e.g. screen-based computer gaming).

The *transduction* of the clapping game from playground to screen is accompanied by a change in modes and interaction. For instance, in the playground version, the player uses both visual and tactile modes to make contact with the hands of the other player (in some “eyes-closed” clapping games touch only is used), whereas in the screen version the tactile mode is omitted and the visual mode therefore becomes more emphasised. This has implications both for the design of the interface

and for the “reading” of the action or interaction. Another example is the location of the gaze of the player. In the playground version this is towards the other player, but in the screen version, the player’s gaze is towards the screen and in particular directed towards the position of the hands. This brings up interesting questions about the relationship of the player to the on-screen visualisation of play/player and questions as to how one designs a user-experience that is perform different, but intended to be no less satisfactory, than the playground version of the game. The reconfiguring and reshaping of modes affects the experience, reading and meaning of the “transduced” text – and this also has implications for how the rules of the games are affected by the move from playground to computer screen.

Figure 2: Hand positions in clapping games

We also had to think about, at a very fundamental level, what the essence of clapping game is. In terms of hand orientations, there are five main hand orientations when the players clap, as shown in figure 2. These are:

- (i) Palm down, fingers pointing at other player (clapping vertically with other player)
- (ii) Palm up, fingers pointing at other player (clapping vertically with other player)
- (iii) Palm facing other player, fingers up (clapping horizontally with other player)
- (iv) Palm to side, thumb up, fingers pointing at other player (clapping horizontally with self)
- (v) Palm to side, fingers up, thumb pointing at self (clapping diagonally with other player)

The repertoire of moves used in a clapping game is, however, far higher. Firstly, these are just end positions – the orientation of the hand between claps is important as it indicates the next action. Secondly, there is non-clapping contact between

players – for instance, in one of the schools studied in the project, there was what we referred to as a “three way clap” where the players place the backs of their left hands together while they perform a sequence of three claps with their right hand. Thirdly, there are gestures in clapping games which don’t involve clapping, but are drawn instead from dance routines or from elaborate handshakes.

5. CHOICE OF TECHNOLOGY

Computer game hardware offers a number of benefits which were felt to be highly applicable to the aims of this project. Firstly, it offers an extremely high price to performance ratio (low price, high performance). As game controllers are produced in such large quantities, their price is substantially lower than the cost of buying its individual components separately. In addition, computer game hardware is extremely robust and widely available.

Developing the Game Catcher involved finding both the position of the hands in 3D space and their orientation. These have to be done with enough accuracy and resolution to enable them to be used to produce accurate animations (both at the time of recording and during playback) and to generate meaningful and useful data. The system has to be robust, and not susceptible to background noise which would show itself as a random “juddering” of the hands. In addition, all of this has to be done at a sufficient frame rate – and with sufficiently low latency – to allow the application to feel responsive to the user.

Figure 3: Structure of main Game Catcher modes

During the course of developing the Game Catcher, we used a number of different solutions to before adopting a “best of breeds” approach which used the Kinect sensor to track hand position and Wiimote controllers to track hand orientation. But

even once we had settled on this combination, we still experimented with a number of different libraries and coding techniques to interface the Kinect with Processing. It is useful, therefore, to discuss briefly the strengths and weaknesses of each of these approaches for the benefit of the wider community.

Video tracking was eliminated very quickly because of the widely-known shortcomings of this approach, which the authors had previous practical experience of, having used applications such as BigEye and others. Our main concern was that video-based tracking is not robust and is too easily affected by outside conditions such as the brightness and colour temperature of the lighting in the room where it is being used or the colour of the clothing worn by the person being tracked. This rendered it unsuitable for the Game Catcher as we wanted a system that could be taken to schools and provide reliable and robust tracking in a variety of locations without a lengthy calibration and setup procedure. We were also concerned about potential speed and framerate issues with video tracking.

Because of these known issues with video tracking, we rapidly switched to an approach which used infra red LEDs, rather than visible spectrum light, particularly as this allowed us to also exploit the strengths of the Wiimote. The Wiimote is normally used with a sensor bar which sits just under (or just over) the television. This sensor bar is, in fact, not a sensor – it has no sensor functionality and actually contains just a set of infra red lights. The lights are used by the Wiimote, which has a camera in its tip, to more accurately measure the orientation of the Wiimote when it is being used to point at or select items on the screen.

We attached an infra red LED to the Wiimotes in the player's hands and used a third Wiimote as a camera pointing at the player to track the position of these LEDs. The advantage of this approach is that it is very fast, accurate and robust, being several times faster than video tracking and also more accurate.

The speed comes from the fact that the Wiimote has a dedicated built-in chip which is optimised to do this image analysis in hardware. Accuracy is enhanced because although the Wiimote has a relatively low resolution camera, the image is interpolated as part of the analysis process to give a much higher effective resolution.

Being infra red, the tracking is unaffected by lighting conditions (providing it is not pointing at a bright, hot, light source such as a bulb or a candle). This means that it is much more reliable and robust than tracking a visible colour. One negative aspect

of this system is that because it is tracking a point source, it can only track in two dimensions (the XY plane). With video tracking, it is possible to use the apparent size of an object to calculate depth (as objects appear larger the closer they are). This works best if one is tracking a sphere (as is the case with the Sony Move controller, which uses this technique). If one is tracking a non-spherical object (as one would be if tracking, say, a coloured glove), the apparent size and shape of the tracked object can change as its orientation changes. This can affect the accuracy and realism of the tracking: changes in the size can affect the apparent distance to the object, while changes in its shape can affect the centre point, and thereby its apparent position.

Researchers at the University of Cambridge have, however, demonstrated that it is possible to track the position of an infra red LED in 3D space using a pair of Wiimotes, by triangulating its position (Hay et al 2008). We were therefore confident that we could, if necessary, adopt the same approach.

In the end, the release of the Kinect – and the fact that it was hacked on its first day of release – rendered this unnecessary. The OpenKinect project's *libfreenect* drivers gave very high frame rates, but did not provide any built-in functions for performing the hand tracking as it only gave access to the depth map generated by the Kinect. As a result, it was necessary to write bespoke code which would track the hands.

A substantial amount of time was devoted to this, but the difficulty in writing routines which were sufficiently resilient to issues such as occlusion meant that we switched to OpenNI once it became possible to access it in Processing (the Java-based programming language used to develop the Game Catcher). This was achieved first by accessing it through the OSC protocol, and then more directly using the Simple-OpenNI library.

OpenNI provided functions to track the whole body and could also track multiple users, providing a persistent ID for each. This led us to expand our work on the Game Catcher and to develop a second version which was capable of tracking several users in a larger area and was therefore suitable for recording and archiving the movements of other playground games such as skipping, hopscotch, etc.

This multiplayer version was intensively user tested at the Children's Conference for the project, being used by three groups of fifteen children in three forty-five minute sessions, but following this conference, development effort shifted back to the single user version of the Game Catcher. This was because allowing the user to play against the

recorded version of a clapping game presented distinct challenges that the multiplayer version could not.

Although the Kinect libraries allow us to track the skeleton, they did not give the hand orientation. We did investigate whether one could assume the hand orientation from its direction of movement, but this did not seem to be reliable in every case. For instance, when the hand is moving forward (away from one's body), one can assume that it will be palm out with the fingers up, but if it is moving across the body, it could be in one of several different orientations. This meant that a hybrid technique was necessary, using the Kinect to track the body position and the Wiimotes to track the hand orientation. This proved to be an ideal solution, as it allowed the strengths of each system to be used.

There were a few issues with the Wiimote, which it is appropriate to point out from a technical point of view. The Wiimote does not, on its own, track yaw (rotation about the Y axis). It relies on accessories to do this – either the Sensor Bar, or the Wii Motion Plus (which contains a gyroscope). Neither of these were suitable in this case. Using the Sensor Bar would have required the user to keep their hands pointing at the bar, and was therefore clearly unsuitable for a clapping game which required free movement in all axes. Using the Wiimote with the Wii Motion Plus would add additional bulk and weight which we felt was not appropriate (though we did briefly investigate whether it would be possible to use the Wii Motion Plus without the Wiimote).

A consequence of this was that we could not tell two key hand positions: the palm out, fingers up, position when one is clapping with the other player (position iii in fig. 2); and the similar position with the palm facing sideways used when one is either clapping obliquely with the other player or clapping with oneself (position v in fig. 2). In addition, the Wiimote suffers from a gimbal lock problem when pointed vertically upwards which meant that when the hand was in one of these positions, the rotation could flip uncontrollably by 180°.

These issues were solved by paying attention both to the limits of human movement and to the conventions of the clapping game and using these to provide an additional level of interpretation on the moves. For instance, when the hands are vertical (fingers pointing up) in a clapping game, it is unlikely that the player's palms are facing their body. Likewise, when the hands are clapping obliquely with the other player (palm sideways as in position v in fig. 2), they will have gone through different intermediate positions than when they are doing the normal palm out clap (position iii in fig. 2).

These rules are used to make the hand “snap” to certain hand orientations (though it should be noted that this only affects the display of the hand as the text file records the raw orientation data).

The theoretical performance of the Game Catcher is shown in table 1. The presence of unavoidable system noise in the depth map reduces these figures slightly in practice, but XYZ accuracy still remains well within acceptable levels. The accuracy with which depth is measured varies, and when an object is far from the Kinect, its movement is measured in larger steps than when it is close. The figure for orientation is the raw measurement and in some hand orientations the hand will snap to 90°. As mentioned above, this orientation measurement is obtained using the built-in accelerometer, not the Wii Motion Plus accessory.

Table 1: Theoretical performance of Game Catcher

Dimension	Performance	Notes
XY position	3mm	Using Kinect
Z position	1cm	Using Kinect
Z range	1.2m to 3.5m	
Orientation	<1°	Using Wiimote accelerometer only (not Wii Motion Plus)

6. VISUALISATION AND ANALYSIS

As the player records a clapping game, two files are generated: a plain text file containing the movement data and an audio file containing the associated sound recording. These files are used to provide the movement and sound when the clapping game is played back. A third file containing the words of the song can be created manually, if required. If this file is present, it will be used alongside the other two when the game is played to display the words of the song on screen with a “bouncing ball” effect.

In addition to allowing the user to play any previously recorded clapping game, the Game Catcher also provides tools with which to analyse and visualise the game. These use the same movement data files as the game. Currently, only one form of visualisation has been implemented as a proof of concept, but this nonetheless indicates the usefulness of the system and generates further ideas. This visualisation shows a stick figure performing the moves of the clapping game and a pair of lines showing the path taken by the hands throughout the entire game. The display can be toggled to show the figure, the paths, or both together (the latter being the default). The movement itself can be played, rewound and

paused at will and in addition, the scene can be rotated in all axes and viewed from any angle.

Figure 4: Analysis mode showing paths taken by hands

Showing the path of the hands throughout the game provides a useful visual summary and enables the most predominant moves from it to be recognised at a glance. We believe that by comparing two images side by side, it would be possible to recognise related clapping routines with similar moves, even though they may have different clapping rhymes. We have not had the opportunity to test this yet, but if it works, it would be a significant breakthrough. Currently, this type of comparison would involve watching two videos side by side and to be able to identify similarities, the videos would have to be shot from similar angles/distances and feature clapping rhymes of similar length performed at a similar tempo (otherwise they would slip out of sync with one another). These difficulties explain why the tracing of variation in clapping games has tended to focus on the words, rather than the movements.

Another way in which these similarities could be identified would be through superimposing stick figures from two different recordings. We believe that this type of visualisation would be most useful in detecting subtle difference and variation, such as that which might occur in a particular clapping game in a particular location over time.

As the movement has been translated into numerical data, it is possible for it to be analysed automatically by computer using statistical analysis and artificial intelligence (e.g. hidden Markov models) to identify similar gestures, patterns or rhythms. Simpler forms of computer-based analysis would also be worthwhile. It would, for example, be relatively straightforward to identify clapping rhythms using simple arithmetic and trigonometry, as the claps can be detected by sudden changes in velocity and direction of movement (during the

game itself they are identified by proximity as this allows us to, in effect, recognise a clap before it occurs).

7. CONCLUSIONS

The development of the Game Catcher prototypes (the single player and multiplayer versions) has proven that a low cost motion capture system built around videogame hardware (Kinect and Wiimote) is both (a) technically viable and (b) useful in practice as a tool for recording, archiving and analysing movement. This setup provides tracking which is both very precise and highly robust under a variety of conditions.

A viable data format has been developed which allows for the potential of players in the multiplayer version to appear and disappear (as they are picked up by the tracking, or lost as they disappear out of frame). This data format is also suitable for the single player version of the Game Catcher. We have provided, in the Game Catcher, a sample of the ways in which this data can be visualised. This is useful in itself, and also suggests further enhancements and alternative uses.

With regard to further development, the size and shape of the Wiimote is a slight issue and we are currently investigating ways to miniaturise the functionality provided by this controller. We envision that a Seeeduino Film offers the most viable solution (probably communicating with the PC via Xbee rather than Bluetooth). This should provide a solution with minimal weight which will fit on the back of a child's hand. We also intend to consolidate and merge the codebases of the single player and multiplayer versions of the Game Catcher so that it can provide a greater degree of flexibility to the researcher and allow them to change from one mode of use to another without switching applications. We are also looking at providing a robust "solution-in-a-suitcase" so that the system can be easily used in the field, both indoors and outdoors.

8. REFERENCES

- Burn, A, Marsh, J, Mitchell, G, Robinson, J, Willett, R. (2011) *Children's Playground Games and Songs in the New Media Age 2009–2011 Project Report*. http://projects.beyondtext.ac.uk/playgroundgames/uploads/end_of_project_report.pdf (accessed: 31-03-2011).
- Hay, S., Newman, J., Harle, R. (2008) Optical Tracking Using Commodity Hardware. *IEEE International Symposium on Mixed and Augmented Reality 2008*, Cambridge, 15–18 September 2008.

Kress, G. (2009) *Literacy in the New Media Age*. Routledge, London.

Mitchell, G. (2010a) Porting playground games into a computer game environment: Game Catcher concepts, aims and issues. *Children's Playground Games in the Age of New Media Interim Conference*, IOE/LKL, 25 February 2010. London. http://projects.beyondtext.ac.uk/playgroundgames/uploads/mitchell_playground_interim_copy.pdf (accessed: 31-03-2011).

Mitchell G. (2010b) *The Multimodality of Physical & Virtual Playground Games* Multimodality & Learning Conference, IOE, London 6–7 July 2010.

OpenKinect. http://openkinect.org/wiki/Main_Page (accessed:31-3-2011).

OpenNI. <http://www.openni.org> (accessed: 31-3-2011).

Simple-OpenNI. <http://code.google.com/p/simple-openni/> (accessed:31-3-2011).