Detecting Gesture Force Peaks for Intuitive Interaction

Samuel Jones
Queensland University of Technology
sam@visualeyes.net.au

Zachary Fitz-Walter
Queensland University of Technology
z.fitzwalter@gmail.com

Dian Tjondronegoro
Queensland University of Technology
dian@qut.edu.au

ABSTRACT

With the release of the Nintendo Wii in 2006, the use of haptic force gestures has become a very popular form of input for interactive entertainment. However, current gesture recognition techniques utilised in Nintendo Wii games fall prey to a lack of control when it comes to recognising simple gestures. This paper presents a simple gesture recognition technique called *Peak Testing* which gives greater control over gesture interaction. This recognition technique locates force peaks in continuous force data (provided by a gesture device such as the Wiimote) and then cancels any peaks which are not meant for input. Peak Testing is therefore technically able to identify movements in any direction. This paper applies this recognition technique to control virtual instruments and investigates how users respond to this interaction. The technique is then explored as the basis for a robust way to navigate menus with a simple flick of the wrist. We propose that this flick-form of interaction could be a very intuitive way to navigate Nintendo Wii menus instead of the current pointer techniques implemented.

General Terms

Algorithms, Measurement, Performance, Design, Human Factors

Keywords

Gestures, Gesture Recognition, Games, Interaction, Nintendo, Wii, Wiimote, Accelerometer, Navigation

1. INTRODUCTION

Gesture interaction for games has taken off with the release of the Nintendo Wii video game console in 2006 [11]. The Nintendo Wii supports gestures as a form of input through the inclusion of an accelerometer in each Nintendo Wii game controller, or Wiimote.

As opposed to a button press, the Wii's gestural controller allows specific physical movements to directly correspond to a control. For example, holding and swinging the controller like a sword triggers the game character to swing his sword. Holding the controller horizontally and turning it like a steering wheel can turn the wheels of a virtual car. The controller can also be used as an extension of the body, where users hold the controller in their hand and run in the game by simply running on the spot in real life. These new controls open a whole new world whereby physical actions that players make can be emulated more intuitively and naturally in the game. When the gesture recognition is executed effectively, gesture input could make for a very realistic experience for the player.

Current gesture recognition techniques used in Wii Games involve the use of raw force data from a gesture device to calculate tilt, rotation and amplitude of the controller as triggers for input. These techniques are mostly non-directional with an inability to discern between any intended and unintended peaks of force made by the player. When a force gesture is made, such as a swinging the Wiimote like a tennis racket there is a peak of force in a direction intended by the user to trigger an action, we call this an *intended peak* of force. However when this gesture comes to a stop there is peak of force in the opposite direction caused by the deceleration of the Wiimote, we call this a *rebound peak*. Most games on the Wii cannot determine which of these force peaks are meant as input, so rebound peaks can trigger actions in any games that utlise this type of recognition. This occurs in games such as the very popular *Zelda Twilight Princess* and *Wii Sports*, both released by Nintendo in 2006.

This paper presents a new gesture recognition technique which adds a level of recognition to force gestures made in order to locate when the peaks in the force occur. Once these peaks have been recognised they can be used to calculate intended peaks of force and cancel the effects of rebound force peaks.

This technique was discovered when we explored a simple way to play virtual instruments using force triggers instead of advanced gesture recognition. Without peak recognition, using only force amplitude would repeatedly play the instrument sounds until the force fell below the trigger force. However, we recognised when the peaks in force occurred and this allowed us to set up triggers for each gesture so we could emulate a guitar strum or drum hit to play a sound. However, using this technique meant that only one direction on each axis could be utilised as input. A gesture in the opposite direction would create the rebound peak that would trigger the sound meant for the original gesture. To fix this issue we devised a recognition technique that allowed us to utilise any direction of the axes as input. This is explored in detail later in the paper.

We then investigated this technique and discovered it could provide the grounding as a robust and intuitive way to interact with menus. We focused on using peak recognition in determining directional flicks of the Wiimote to navigate simple generic 2D menu grids commonly used for main menus, text input and option selection in video games. Based on a user evaluation, we found that the current pointer technique employed on the Nintendo Wii can be slow and has usability issues that encompass namely speed, accuracy and screen real estate. Gesture flicks for navigation have the potential to overcome a number of these issues through careful implementation and interface design. With a focus on the key design principles of simplicity and intuition this paper will explore the potential of utilising simple gestures instead of pointer devices such as the IR camera on Wiimote.

2. GESTURE RECOGNITION

2.1 Overview and Brief Background

Biologists broadly define the term "gesture" as all kinds of instances where an individual engages in movements whose communicative intent is paramount, manifested, and openly acknowledged [10]. According to a case study of gestures by Mitra and Acharya [8] gestures are expressive movements with the intent of conveying meaningful information and interacting with the environment. A system that can recognise these gestures and process them as input is a gesture recognition system.

Popular gestures for physical human computer interaction include *body*, *head and facial* and *hand and arm* gestures [7].

2.2 Recognition Techniques

A number of different techniques are available to measure accelerometer based force input received from a handheld gesture device such as the Nintendo Wiimote.

The Wiimote is a cost effective gesture handheld device which extends the body to measure hand and arm gestures. An accelerometer inside the Wiimote measures force values acting on the different axes and in turn, these values provide the basis for computer gesture input and recognition.

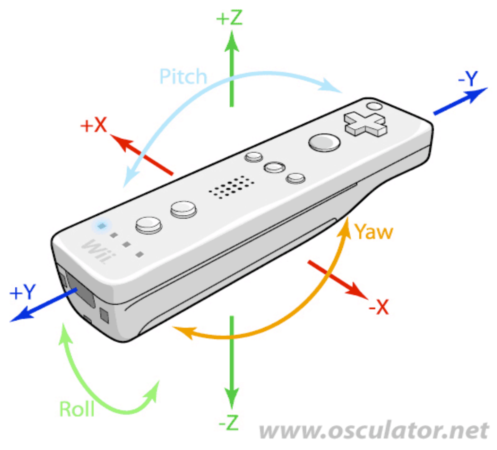

Figure 1. Wiimote displaying different axes and rotations of the accelerometer [15].

According to Mäntyjärvi et al [7] gesture recognition techniques can be separated into three types: 1) *Measure and Control*, 2) *Discrete Gesture Command*, and 3) *Continuous Gesture Command*. Measure and Control is a simple recognition technique which has the lowest complexity. It uses direct measurement of tilting, rotation or amplitude of a gesture device as input. Discrete and continuous gesture command techniques are executed based on the hand movement recognised by the machine and allow for complex gestures to be recognised. With discrete gesture command the start and stop of a gesture needs to be defined, usually with the press and hold of a button while the gesture is carried out. Continuous gesture recognition is instead carried out on a continuous flow of hand force input in real time and is the most complex of the three.

It seems that Mäntyjärvi et al [7] have not considered direct measurement and control systems as gesture recognition systems because generally the operating principle of this technique is to map directly to the control the measurement of tilt, rotation or amplitude. However we hypothesise that this is not entirely true, as the act of defining limits reached by a force amplitude trigger input constitutes basic recognition in itself. For example, when playing *Wii Tennis* a force two times greater than gravity needs to be recognised to swing your tennis racket in the game. This concept of recognising force peaks as triggers is the basis of our peak recognition technique.

3. RELATED WORK

There seems to be a current trend to use the measure and control of gestures as input over complex recognition techniques. Not only do games on the Nintendo Wii console use simple gestures for interaction but other research has explored the use of simple gestures to control games on mobile devices [6].

This trend could be due to a number of facts. First, measure and control recognition is the simplest recognition type to implement. Advanced recognition techniques require the use of a statistical model or an AI system to recognise gestures, whereas simple gestures using measure and control simply require force triggers. Secondly, advanced gesture recognition requires each gesture to be trained [4] and needs to be calibrated to the users' individual gesture variations. Even after all the gestures are trained, there is still the issue of accuracy which all gesture recognition systems face [14].

Previous research into gesture recognition has generally focused more on technical aspects: improving the accuracy and speed of advanced recognition techniques. Advanced gesture recognition has been explored using the Wiimote, with research focusing on the application of an advanced gesture recognition system for the device [14]. However, there has been little study into the usefulness of recognising simple measure and control gestures. Linjama and Kaaresoja [6] have explored the use of simple gestures to control games on mobile devices. Their research entails the use of an accelerometer in a mobile phone where they focused on exploring simple, haptic interaction suitable for mobile devices. While the focus of their research is on the use simplistic gestures in a mobile context, our research instead looks at the use of simple gestures in home interactive entertainment, such as the Wii.

4. PEAK RECOGNITION

4.1 Limitations of Measure and Control

Simple gestures have provided just enough control that most basic video game interaction need. The tilt on the Wiimote has been used to control the steering in *Mario Kart Wii* released by Nintendo in 2008. The rotation of the Wiimote has been used to emulate a key turn in *Zack and Wiki* released by Capcom in 2007 and force amplitude was used to imitate hitting a tennis ball with a racket in *Wii Sports*.

However there are limitations with this interaction, one in particular is the inability to discern between intended and unintended force peaks. We refer to a force peak as a spike in the amplitude of a gesture device caused when force is applied to one or more of the devices' axes. A gesture such as imitating a tennis swing can cause such a force peak. Currently force amplitude is utilised in many games as control input where the user has to apply a certain amount of force in a certain direction to trigger an action. As an example, the Wii game *Zelda: Twilight Princess*, requires the player to apply a certain force on the Z axis of the controller in order to activate the character's shield. When this gesture is performed a spike in force on the Z axis is recorded with a peak in the force appearing when the gesture is made. This is the *intended peak,* as it is made by the gesture intended to trigger an action.

However when a peak like this is made there is always a rebound peak in the opposite direction when the user stops making the gesture. This is the *rebound* or *unintended peak*.

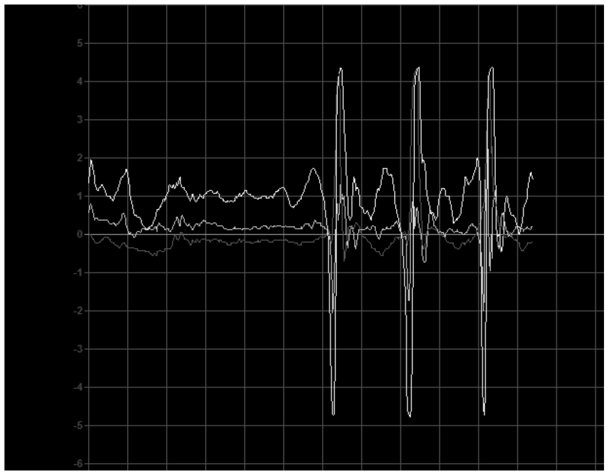

Figure 2. Force input from an accelerometer showing six peaks of force on the Z axis, each intended peak has a rebound peak in the opposite direction (G Force vs. Time).

This means that in *Zelda* the shield can be triggered by the rebound peak if the player performs a gesture in the opposite direction. The action will trigger if the rebound peak has a force that is greater or equal to the force required to trigger this action, which is usually the case as shown in figure 2.

This not only breaks the realistic experience of using gestures but one of the biggest drawbacks is the inability to utilise the opposite force direction as input. Both rebound peaks would trigger opposite actions. This means that the use of simple gestures is limited when compared to advanced gesture recognition, that is, unless some basic recognition is applied to it. Our research has led to the development of a simple recognition algorithm that locates the force peaks as they occur and then identifies if they are intended or rebound peaks.

4.2 Peak Testing

Peak testing is at its simplest, an extension of the measure and control interface type. It adds basic recognition to calculate the amplitude force fluctuations that are evident in a continuous stream of accelerometer data from a gesture device.

When designing input we identify different peak fluctuations to coincide with a user's intention to commit an action. For each axis of an accelerometer, X, Y and Z, we can determine whether the user moved the Wiimote in a positive or negative direction. Therefore we can recognise a left, right, up or down movement by measuring the range of data for positive and negative peaks.

This process records and samples numeric force amplitude data and evaluates when the minimum and maximum values, or peaks, have occurred. This technique is based on interpreting a peak in force values by comparing force values to an average resting position of the handheld gesture device. Normally, a value of zero on all axes would be assumed as the resting value of a gesture device that the user is holding in an idle or non-active position. However, one or more of the axes will always be affected by gravity. Therefore, the average resting force is calculated from the continuous force data streaming from the Wiimote in order to obtain an average resting value at any given point of time. In this way gravity is always taken into account when calculating force peaks.

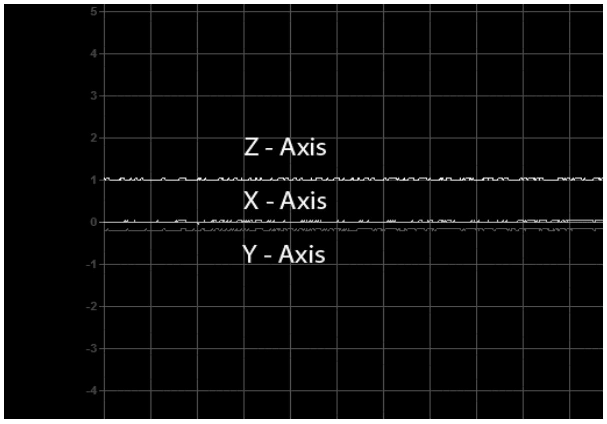

Figure 3. Force acting upon Wiimote at rest with the Z-Axis being affected by gravity (G Force vs. Time).

With this information the average resting position of the Wiimote is calculated and then the start and end of a motion is determined by the rise or fall of values from the average resting position. Continuous data can then be segmented when the force passes a trigger value and then returns once again to the average resting position. This segmented data provides peaks that can be used to set off actions which can be customised by defining the amount of force required to trigger it.

This simple process can be applied to the Wiimote, or any other device that utilises an accelerometer, to give us the ability to recognise hand flicks made with the controller in any direction. This makes this process the perfect way to interact with a number of different applications.

4.3 Implementing Gesture Instruments

Peak testing was first implemented in an application that allows users to play sounds using Wiimote peak gestures. In this way virtual instruments could be created that are controlled by realistic gestures rather than button presses.

The virtual guitar and drum demonstration we created, named *Wii Jam*, allows for the composition of rock style music to be made with realistic gestures such as strumming and drumming. Many popular music games such as the *Guitar Hero* series released by Activision and the *Rock Band* series released by MTV Games, constrict the player to following pre-recorded songs using realistic physical controllers. There are games such as *Jam Sessions* released by Ubisoft in 2007 on the handheld console, the Nintendo DS, which allow for music composition however there are few mainstream games that let you compose your own music using realistic actions. This mainstream composition was the driving force behind the development of the *Wii Jam* application.

Peak gestures provided an intuitive way for virtual instruments to be controlled. In the first version, *Wii Jam* supported two instruments – the Guitar and the Drum. The guitar was controlled by strumming the Wiimote in a way which emulated the strumming of a real guitar. When a strum produced a force peak over the trigger force value a guitar chord was played. The drums utilised two Wiimotes and when held like drumsticks and swung to produce force peaks, drum sounds played.

Figure 4. Wii Jam - virtual instruments using force gestures

To trigger a predetermined sound we simply defined a gesture which translates to a specific axis force peak. For example, the drums used peak testing to recognise when a negative force on the X axis occurred, as this provided a similar gesture to that of hitting a snare drum.

4.4 Peak Gestures in Menu Interaction

After implementing *Wii Jam* successfully using peak testing, we discovered that this same technique could lead to a novel way to navigate basic menus. We believe that current pointer techniques employed using the Wiimote infrared camera tend to be slow and subject to usability issues which include slow item selection speed and lack of screen real estate.

The alphabet selection screen in particular on the Nintendo Wii Internet Browser requires precise pointing and a stable hand to select the individual letters for input. From experience, pointing with the Wiimote can become tiresome when long input is required, such as the entering of web addresses.

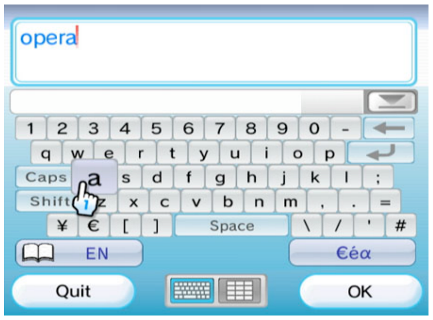

Figure 5. Character input screen for Opera browser on the Nintendo Wii [11].

Recognising hand flicks on a two dimensional plane could provide an alternative to using the pointer technology. When combined with an efficient keyboard layout this technique could provide an experience that is not only more comfortable and accurate, but also quicker. We discovered that wrist flick recognition could be implemented using peak testing where navigating a two dimensional grid menu would require a flick from the Wiimote in either an up, down, left or right direction. The benefits of using gesture flicks over pointers encompass a greater use of screen real estate, accuracy and speed. Interaction is the same wherever the menu is placed on the screen, accuracy is the same for whatever sized menu items are used and menu access is reduced from a two step point and click to a one step flick.

With pointing devices, speed and accuracy depends on a number of factors that include button size, distance and placement. Fitts' law provides an excellent can be applied to computer interface design as it predicts human movement and motion based on time and distance [1]. Fitts' law still applies to gesture flicks but in a slightly varied way. The distance and size of menu items does not affect the accuracy or speed of the gesture flicks like it does pointing devices. Instead, speed is based on the user's reaction time while accuracy depends on how precisely the gesture is executed and recognised.

Therefore, to outperform current pointer techniques, the focus of gesture flicks is to correctly interpret what gesture the user intends to make, as well as designing an intuitive interface that encourages quick user response.

The QWERTY keyboard layout illustrated in figure 5 is not a feasible layout for gesture flicks, as to get from letter Q to letter M would take at least 9 flicks. Instead gesture flicks could work better on a circular menu made for a movement in any direction on a two-dimensional plane. For example a modified T9 layout that is employed for mobile phone users [13] with predictive text could work successfully. Using a layout like this, a quick flick in the desired direction with the Wiimote is all that is needed to select a letter rather than pointing at the character and then pressing a button to select it.

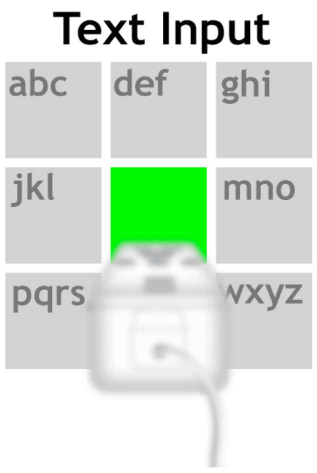

Figure 6. A proposed layout for quick text input using gesture flicks. Inspired by the T9 predictive text layout for mobile phones [13].

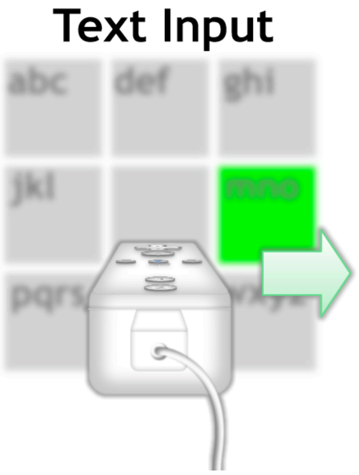

Figure 7. Flicking the Wiimote to the right would choose the "mno" selection.

The third axis could even be utilised for players to push and pull through three dimensional menus. As force flicks could theoretically be detected in any direction this opens up a world of possibilities for interaction.

5. DETECTING GESTURE PEAKS

5.1 Calculating peaks

To make use of the Wiimote's internal accelerometer in application interfaces we must analyse and calculate the significance of fluctuations evident in a continuous stream of accelerometer data.

The peaks of these fluctuations coincide with a user's intention to commit an action. For each axis of an accelerometer, X, Y and Z, we can determine if the user moved the Wiimote in a positive or negative direction. This can then be interpreted as a left, right, up or down movement by measuring the range of data for positive and negative peaks. The programmatic steps for calculating positive and negative peaks are shown in figure 8.

```
add current force to history
if current force is equal to or very close to the
avg resting value (we're resting or movement has
ended)
       attempt to find peaks
       clear history
else
       reset peaks (movement in progress, no valid
peaks)
if history length is long enough
       calculate average resting position
```

Figure 8. Peak Testing Pseudo Code

To measure the range of data for positive and negative peaks we must first record sets of historical data. Each set of data must be short enough to be parsed immediately in order to provide real time input of each gesture performed. Therefore, we must decide when the user started and stopped their motion for each axis.

To evaluate when the user has started and stopped a motion we need to find the resting position of the gesture device held by the user. When we consider the data ascertained from the Wiimote we find that each value in the dataset lies between a certain force range which can be positive or negative depending on the axis. Therefore, a value that resides in the middle of the range could be a marked as a suitable resting position of the Wiimote. Normally a value of zero would then be assumed to be a point where the user is in an idle or non-active position. This is true only when the axis being analysed is not lying parallel to the Earth's surface and the user's velocity in that direction was at zero to begin with.

With this information we can calculate the average resting position of the Wiimote. We can then determine the start and end of a motion by the rise or fall of force values from the average resting position. After establishing how to segment this continuous data earlier, we can see that determining peak values is a very simple process.

5.2 Cancelling Rebound Peaks

By using peak testing we can discern between intended and unintended force gestures made by the user. This gives us a robust recognition system that can identify in which direction the Wiimote is being moved by cancelling any unintended peaks that would trigger input. One of these unintended peaks would be a rebound force peak made by a gesture on the opposite axis.

To cancel the rebound peak the recognition system simply infers that a rebound peak will arrive in an expected time after an intended peak is made. In this way it ignores the next peak in this given time slot. If this rebound peak doesn't arrive within the time it is assumed that a rebound peak did not occur and therefore it is not necessary to cancel the next peak.

5.3 Defining Peak Testing

In a publication on gesture interaction, Mäntyjärvi et al [7] define a table of gesture recognition types. In a way, Peak Recognition is an extension of Measure & Control and could be inserted into the table in the following way.

Table 1. Gesture recognition properties of movement sensor based user interfaces, updated to include Peak Testing

Interface type	Operating principle	Customisation	Complexity
1. Measure & control	Direct measurement of tilting, rotation, or amplitude	-	Very low
2. Peak Testing	**Peak recognition of amplitude with rebound peak cancellation**	**Limited Customisation**	**Low**
2. Discrete gesture command	Gesture recognition	Machine learning, freely customisable	High
3. Continuous gesture command	Continuous gesture recognition	Machine learning, freely customisable	Very high

Peak Testing can be seen as providing a stronger link between the basic measure and control technique and the more advanced gesture recognition techniques. It provides more customisation than measure and control in than the way that it can discern between intended and rebound force gestures. Also, the complexity of this technique is kept relatively simple thanks to its straightforward algorithm.

6. EVALUATION

To evaluate the possibilities of this technique, two simple tests were developed that explored the feasibility, usability and technical aspects involved with implementing this recognition technique. The first test used *Wii Jam* to explore how users responded to interacting with peak testing and to see if it provided an engaging system of control. The second test explored the advantages and disadvantages of using peak testing to navigate a basic menu and to see how it could be compared to the current pointing techniques used on the Nintendo Wii.

6.1 General Respondent Information

The Nintendo Wii is marketed towards an expanded audience with the official web site stating that the Wii "gives parents and grandparents a chance to play games with their children" and "it gives gamers and traditional non-gamers a chance to share the same experiences in this new generation of gaming" [12]. In this way the sample of twelve users who participated in both tests had ages that ranged between 14 and 86 years. Of the twelve only one had never used a Nintendo Wiimote before and only three of the twelve actually owned a Nintendo Wii console. More than half of those who didn't own a Wii spent at least one to five hours playing video games every week.

The participants' favourite games were noted to include *Mario Kart Wii, Zelda Twilight Princess, Guitar Hero III* and *Super Smash Bros Brawl*. All of these games, except for one, utilise simple force amplitude measurement as a form of input for control.

6.2 Wii Jam Test

6.2.1 Outline

In this first test participants experimented with the two virtual instruments implemented in *Wii Jam* using peak testing. The first instrument, a virtual guitar, required a two stage process of playing a sound. First the users picked a chord using the joystick and then made a strumming gesture with the controller to trigger the sound. The drums on the other hand were simply triggered with a drum-hitting gesture. The aim of this test was to introduce the users to peak testing and record their initial response to the interaction method. The secondary purpose was to record a response comparing the one step gesture required to trigger the drum to the two step gesture required for the guitar.

6.2.2 Results

After testing the virtual guitar and drums participants were asked to scale both instruments against three factors for a basic response. These included whether the virtual instruments were easy to use, whether the interaction was engaging and realistic and if interaction was enjoyable. All but one respondent strongly agreed or agreed that both the guitar and drums were easy to use, engaging and fun.

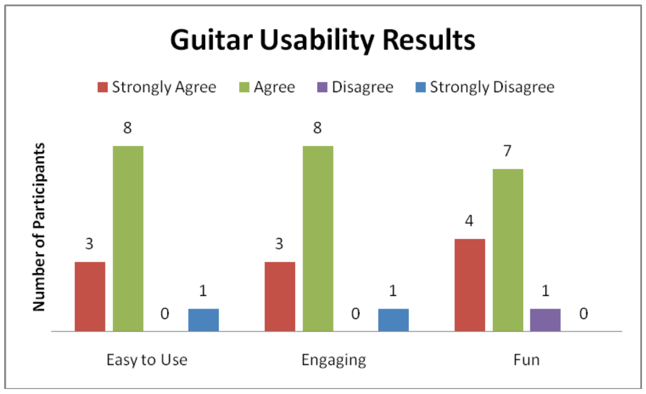

Figure 9. Comparison of guitar usability results.

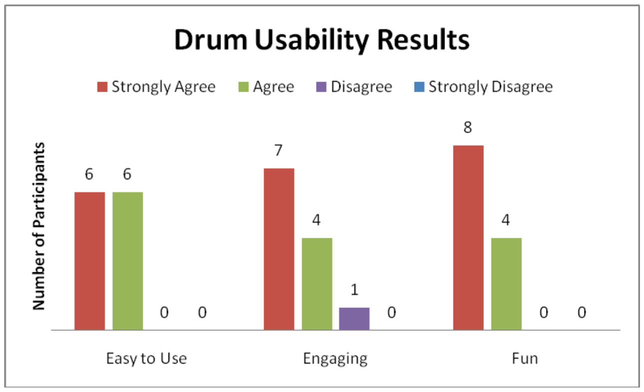

Figure 10. Comparison of drum usability results.

When the two instruments were compared to each other, the drums rated higher than the guitar in each area. Based on observations, we are led to believe that this occurred because the drum interaction was simpler to that required by the guitar.

6.3 Menu Test

6.3.1 Outline

The second test utilised peak testing with some rebound peak cancellation to investigate accuracy and speed of navigating a basic two item menu with gesture flicks and a pointing device. The purpose of this test was to investigate if initial results seemed positive for further investigation into providing a better alternative to the current pointer implementation found on the Nintendo Wii.

This test's design draws from a web based test of Fitts' law [5] that presents two different coloured menu items to the user and requires them to click on one of them. Each time the size and distance of the items changes in order to see how these aspects affect the accuracy and speed of the user. In a similar way our test employed two different coloured menu items that changed in size and distance after each correct selection. As the method for cancelling rebound peaks was still being developed, the menu was limited to two items which required either a left or right flick to be activated with gestures.

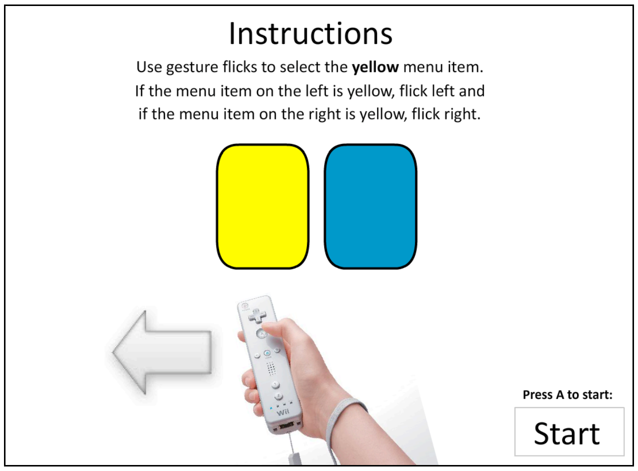
Instructions
Use gesture flicks to select the **yellow** menu item.
If the menu item on the left is yellow, flick left and
if the menu item on the right is yellow, flick right.
Press A to start:
Start

Figure 11. Gesture flick user test instructions.

The pointer part of the test was completed using the Wiimote's inbuilt infrared camera as a pointer, like in current Wii games.

For both gesture flicks and pointing, a total of ten menu selections had to be completed by each participant with the best of three attempts by each participant recorded. With each correct selection, the menu layout would change with the menu items scaling and moving away from each other. The participant's overall selection time and accuracy was recorded.

6.3.2 Results

It was found that generally participants completed the menu test a little faster with gesture flicks than with pointing.

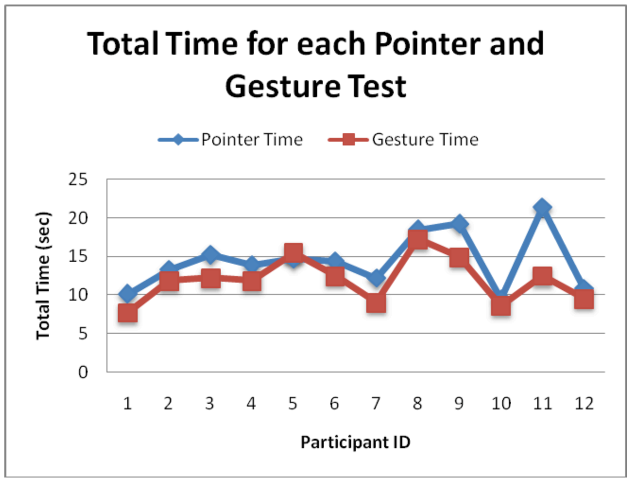

Figure 12. Comparison of test completion time in seconds.

On average with the two item menu, selection speed with gesture flicks was 2.48 seconds faster than pointing selection. Gesture flick accuracy was slightly higher than pointing accuracy where participants missed on average 0.75 menu items out of ten items with gesture flicks. With pointing an average of 0.92 menu items were missed. This gave gesture flicks an accuracy of 92.50% while pointing had only a marginally lower accuracy of 90.08%.

When asked after the experiment which interaction was the preferred, mostly participants chose gestures over pointing to navigate the menu items. All but one participant said they would prefer gestures over pointing if it could be made more accurate.

Overall we found that gesture flicks performed well in all areas for basic menu navigation. We believe these results are a positive start and provide encouragement to further explore the use of directional gesture flicks for interacting with menus.

7. CONCLUSION

Gesture input on the Nintendo Wii has provided a very different approach in the way games can be controlled. Our exploration into using force gestures to emulate instruments can be seen as having positive results from the test participants, with most agreeing that they were easy to use, realistic and enjoyable.

The peak testing method shows promise in providing an intuitive recognition technique that allows for greater customisation of input from gesture devices. The greatest advantage of our proposed technique is the ability to pick up intended force input in any direction and cancel rebound peaks. This technique is simple and we urge its application in future games to make for a more realistic experience where unintended rebound peaks have no affect on the interaction.

We found that our proposed technique also lends itself to becoming a very robust and intuitive way to navigate selection menus as shown in the tests.

7.1 Future Work

Future work would look to refine the recognition technique and test the boundaries of gesture flicks, looking to determine the optimal number of menu items that could be accessed in a circular menu. Wii Jam could also be explored further, with ambitions to port it across to the Wii itself now that Nintendo have opened a small developer's channel.

8. ACKNOWLEDGEMENTS

We would like to acknowledge the other fantastic Wii Jam group members; Peter Riesz and Craig Stewart as well as Dr. Margot Brereton for her support and input to the project.

9. REFERENCES

[1] Amento, B., Brooks, P., Harley, H., & McGee, M. (1996) *Fitts' Law*. Available from: http://ei.cs.vt.edu/~cs5724/g1/#summary. Accessed 1 July, 2008.

[2] CBS News (2005) *Gesture Glove Not Science Fiction*. Available from http://www.cbsnews.com/stories/2005/08/23/eveningnews/main792311.shtml. Accessed 1 July, 2008

[3] Cohen, C. (1999) *A Brief Overview of Gesture Recognition*. Available from: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/COHEN/gesture_overview.html. Accessed 1 July, 2008.

[4] Eickeler, S., Kosmala, A., & Rigoll, G. (1998) Hidden markov model based continuous online gesture recognition. In *Proceedings of the 14th International Conference on Pattern Recognition.* (pp. 1206-1208). Washington, DC, USA: IEEE Computer Society.

[5] Goldberg, K. (2008) *A Web-based Test of Fitts' Law*. Available from http://www.tele-actor.net/fitts/index.html. Accessed 1 July, 2008.

[6] Linjama, J., & Kaaresoja, T. (2004) Novel, minimalist haptic gesture interaction for mobile devices. In *Proceedings of the*

Third Nordic Conference on Human-Computer Interaction. (pp. 457-456). New York, NY, USA: ACM.

[7] Mäntyjärvi, J., Kela, J., Panu, K., & Sanna, K. (2004) Enabling fast and effortless customization in accelerometer based gesture interaction. In *Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia.* (pp. 25-31). New York, NY, USA: ACM.

[8] Mitra, S., & Acharya, T. (2007) Gesture recognition: a survey. *Systems, man, and cybernetics, Part C: Applications and Reviews, IEE, 37*(3), 311-324.

[9] Myers, B (1998) A brief history of human computer interaction technology.*ACM interactions, 5*(2), 44-54.

[10] Nespoulous, J., Perron, P., and Lecours, A. *The Biological Foundations of Gestures: Motor and Semiotic Aspects*. Lawrence Erlbaum Associates, Hillsdale, MJ, 1986.

[11] Nintendo. (2006) *The Nintendo Wii*. Available from http://www.wii.com. Accessed 1 July, 2008.

[12] Nintendo. (2006) *Wii Experience.* Available from http://wiiportal.nintendo-europe.com/14.html. Accessed 1 July, 2008.

[13] Nuance Communications. (2007) *T9 Text Input.* Available from http://www.t9.com/. Accessed 1 July, 2008.

[14] Schlömer, T., Poppinga, B., Henze, N., & Boll, S. (2008) Gesture recognition with a wii controller. *Proceedings of the Second International Conference on Tangible and Embedded Interaction.* (pp. 11-14). Bonn, Germany: ACM.

[15] Troillard, C. (2008) *Osculator FAQ.* Available from http://www. osculator.net/wiki/Main/FAQ. Accessed 1 July, 2008.