# Polyrhythm Hero: A multimodal polyrhythm training game for mobile phones

John K. McNulty

Sonic Arts Research Centre
Queen's University Belfast
Belfast BT7 1NN
Northern Ireland, UK
+44 (0)28 90974829
jmcnulty05@qub.ac.uk

**Advisor:** Dr. Sile O'Modhrain

**Abstract**

This paper describes the development of Polyrhythm Hero, an iPhone game that explores the impact of haptic, audio, and visual modalities on the learning of complex rhythmic relationships. First, the game is explained, along with the motivating factors behind its creation. Next, a background in both polyrhythm pedagogy and similar mobile music applications is given to provide a context for this work. The layout of the application is then covered, detailing the functionality of the settings mode, training mode, and gaming mode. Next, the evolution of the technical design is covered, with particular attention to audio, visual, and haptic programming decisions. Finally, the results of a quantitative and qualitative study on the game's effectiveness as a rhythm-training tool are presented. The paper concludes with a discussion of the implications of the experiment, along with suggestions for future work.

# 1. Introduction

"Polyrhythm Hero" is a new mobile rhythm training game that challenges users to tap the two rhythms of a polyrhythm, given a combination of audio, visual, and haptic cues. A screenshot showing the main view is shown in Figure 1. In this game, the player is presented with two rhythms simultaneously. Each rhythm represents an opposing subdivision of time but both eventually resolve to a common downbeat. As these rhythms play the user is required to tap the LEFT button in time with the first rhythm and also tap the RIGHT button in time with the second rhythm. The user's score reflects how accurately this task is performed. The example in Figure 1 shows a 4 against 3 polyrhythm, which means that in one measure of music the left hand will be tapping quarter notes while the right hand must tap whole note triplets. At the player's discretion, the task of polyrhythm tapping can be aided by the following modalities of feedback:

| Mode | Description |
|---|---|
| Audio | The beats of Rhythm 1 can be made to trigger a snare drum sample, if desired. The beats of Rhythm 2 can be made to trigger a ride cymbal, if desired. |
| Visual | A static visual can be shown for each rhythm that helps to convey note lengths by using line segment lengths. |
| Haptic Vibration | The iPhone can be made to vibrate on the downbeat that both rhythms share. |

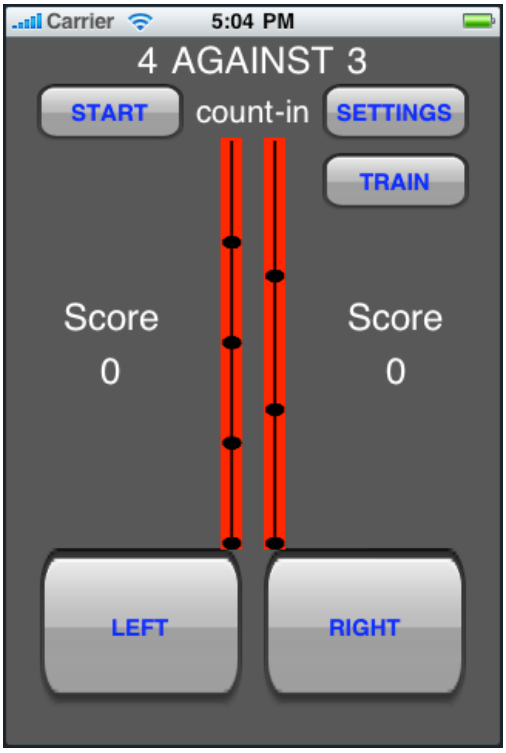



**Figure 1. Main View Screenshot**

Polyrhythm Hero shares commonalities with other rhythm trainers but it differentiates itself in several ways. First, it allows the user to set the exact combination of feedback modalities. Second, it uses an optional haptic vibration to help users identify the first beat in every measure. Third, the software allows the user to train on any N against M polyrhythm, where N and M can range between 1 and 16. Finally, it uses a unique static segmented line, similar to a piano roll, to visually convey the relationship between the two rhythms.

## 2. Design Motivation

The initial inspiration for this design came from attending a jazz clinic in which the clinician used accented clapping and vocal utterances to teach the audience to count polyrhythmic music. The clinician wanted to demonstrate how both rhythms in a polyrhythm would periodically share a common downbeat. Because of the amount of information the clinician needed to convey and the complex nature of that information, it seemed a natural fit to convey this information in a multimodal manner.

Initially, the goal of this project was to create an application that would clearly illustrate the concept of a periodic shared downbeat in a polyrhythm through use of audio, video, and haptic cues. The iPhone was chosen for its ability to provide all of this in a very portable footprint. Moreover, the multi-touch screen promised a very flexible interaction interface. The initial vision was to have the user tap the screen on the shared downbeat, a task that would hopefully reinforce the phrasing of the polyrhythm.

Once the original vision was realized, however, initial testing made it obvious that simple downbeat recognition was neither challenging nor particularly instructive. Further research into polyrhythm pedagogy led to the idea of tapping every beat in both rhythms that made up the polyrhythm. This method can be attributed directly to "An Easy Method for Understanding and Playing Polyrhythms" [9], a recent article on using tapping to practice polyrhythm for piano.

## 3. Background

Polyrhythm Hero takes inspiration from previous work in complex rhythm pedagogy and other mobile music applications. This section will examine works that have similar elements, so as to provide a clear context for the game.

### 3.1 Complex Rhythm Pedagogy

Listening to aural examples is common way to introduce the concept of polyrhythm. The more difficult task of actually playing a polyrhythm is typically taken one rhythm at a time. While instruments such as piano and percussion require that a single musician play both rhythms simultaneously, instruments such as the flute or clarinet cannot play two rhythms simultaneously and thus need accompaniment for polyrhythm to occur.

In the classroom, a common way to introduce polyrhythm is to break it into its two component rhythms. One half of the class learns to clap the first rhythm and the other half of the class learns to clap the second. When the entire class is made to clap at the same time, a polyrhythm is heard.

Learning to play both rhythms in a polyrhythm is a difficult task. Pedagogical approaches to this task include walking in one time while clapping in another [1] and tapping one rhythm with the left hand while tapping the second rhythm with the right.

## 3.2 Mobile Music Applications

Gillian's Scratch-Off [2] application is one of the most relevant examples of a multimodal music application for mobile devices. Unlike Polyrhythm Hero, it does not deal with complex rhythms nor is it meant to be a pure rhythm trainer. However, it does score players on their ability to perform gestures in time with music and it does explore how combinations of feedback modalities influence game play. Other research involving multimodality on mobile devices include SmartKom Mobile [3], The Mona Project [4] and AmbiLearn [5]. This software also takes cues from the iPhone Ocarina [6], as both are music generating mobile applications that rely on real-time input from the multi-touch screen.

Tapping in time to music is a common theme in many mobile applications. Rhythm Heaven and Rhythm Paradise are rhythm tapping games on the Nintendo DS platform. Tap Tap Revolution for iPhone is a prime example of a mobile tapping game that happens to also use visual animations that are derivative of games like Guitar Hero and Rock Band. Although Polyrhythm Hero does not currently use any visual animations, the static line segment graphics of the game are read from bottom to top and vertical distance represents time. This is a paradigm that was popularized by Guitar Hero and its descendants.

Possibly the most closely related mobile rhythm training game is the Dolejsky application, "Rhythm for iPhone" [8]. Both applications are similar in that the user is asked to tap a rhythm and their score reflects their tapping accuracy. Unlike Polyrhythm Hero, Dolejsky uses traditional music notation and uses only one button where the user taps just a single rhythm. No haptic feedback is available and the visual score cannot be turned on or off. John Ferland's "Rhythm In Reach" for iPhone is a very similar product, based on a single tap button and traditional music notation.

One final comparison might be "Drums Challenge" for iPhone. This application is similar to Polyrhythm Hero in that it scores based on tap accuracy, teaches rhythm aurally, and requires users to tap more than one rhythm simultaneously. However, it uses visual animations, it does not specifically train in polyrhythm, and has modalities and tempos which are not configurable.

## 4. Software Description

Functionally, this software can be divided into a settings mode, a training mode, and a gaming mode. Unless otherwise specified, changes made in the settings mode apply to both the training mode and the gaming mode. These modes of operation are covered in detail in the following subsections.

### 4.1 Settings Mode

Pressing the SETTINGS button on the main game screen takes the user to the Settings screen shown in Figure 2. For convenience, all settings on this screen are stored even when the application is exited. Consequently, at application startup, all settings are just as they were at the end of the previous session.

The majority of the settings screen is broken into two columns, with settings for Rhythm 1 on the left and settings for Rhythm 2 on the right. The following is a detailed description of the settings that can be turned on or off independently for each rhythm:

A. Spoken switch – when in the ON position, a human voice is heard counting the beats of the rhythm

B. Tick-tock switch – when in the ON position, a metronome tick tock sound is heard counting the beats of the rhythm (the tock sound on beat 1 and the tick sound on all other beats)

C. Visual switch – when in the ON position, the display shows the segmented lines which visually represent the note durations

D. Audio switch – when in the ON position, an audio sample is heard on the beats of the rhythm (a snare drum sample for Rhythm 1 and a ride cymbal sample for Rhythm 2)

E. Balance slider – allows for stereo panning of all sounds for a given rhythm

F. Subdivisions slider – determines how many subdivisions the measure will be broken up into (ONLY applies to training mode, NOT gaming mode)

Finally, there are three settings that apply to both rhythms. These settings are:

A. Tempo slider – sets the beats per minute of the more frequently occurring rhythm (so in a 4 against 7 polyrhythm the tempo slider would be setting the beats per minute for the rhythm with 7 subdivisions)

B. Vibration switch – when in the ON position, the iPhone vibrates for 400ms starting on the downbeat that is shared by both rhythms. This switch has no effect on the iPod Touch.

C. Measures Per Round textfield – determines the number of measures in a round. In the training mode, this is the number of measures of practice that are desired. In the gaming mode, this is the number of measures that a user must play of a given polyrhythm before being evaluated.

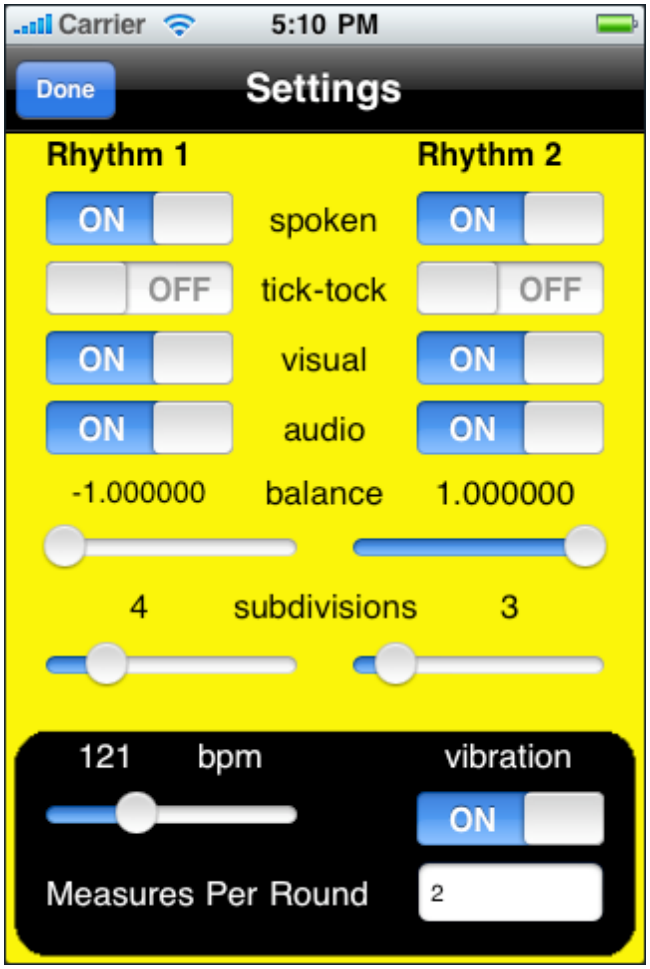



**Figure 2. Settings View Screenshot**

## 4.1 Training Mode

In training mode, the user can practice any polyrhythm they choose. The exact polyrhythm to use in training is specified by the current settings as configured in the settings mode. Upon pressing the TRAIN button on the main game screen, the user would hear a verbal count-in that corresponds to the faster of the two rhythms. In the case of a 7 against 4, the count in would be "1, 2, 3, 4, 5, ready, and". In the case of a 3 against 2, the count in would simply be "1, ready, and". The user may then tap along with the polyrhythm for a number of measures specified in Measures Per Round, as specified in the settings mode. The user's total score for each hand is displayed at the end of the round. The user can press the TRAIN button again to repeat the same training exercise or press the SETTINGS button to go into settings mode and change the training parameters.

## 4.2 Gaming Mode

Gaming mode is very similar to training mode but it ignores the subdivisions specified in the setting mode. Instead, gaming mode starts the user out with a simple 1 against 4 polyrhythm. If the user taps the polyrhythm with enough accuracy, they are automatically moved to the next level. If sufficient accuracy is not achieved, the user must repeat the current level. Sufficient accuracy in gaming mode is currently defined as achieving a score of greater than or equal to half of the maximum possible score for *each* of the two rhythms in the polyrhythm. This forces the user to have adequate accuracy with both hands, as opposed to an averaging scheme where the user's perfect score on one hand might help to hide a substandard score from the other hand.

The game consists of the following ten levels, presented in order of increasingly difficulty: 1 against 4, 2 against 4, 6 against 2, 3 against 6, 3 against 2, 3 against 4, 3 against 5, 5 against 3, 4 against 5, and finally 7 against 4. The game is over when the user has completed all ten levels or ten minutes have elapsed, whichever comes first. Separate scores for the accuracy of each button are recorded to the iPhone for retrieval at the end of the game.

# 5. Technical Implementation

This application required two screens: a main view and a settings view. The most straightforward way to accomplish this was to start with the Utility template provided by Apple, which creates a generic Xcode project with two views. Determining how to best pass information back and forth between the two views was not as straightforward. After careful consideration, using NSUserDefaults was chosen because it allows for simple access to the same variables from different views. Moreover, using NSUserDefaults had the added benefit of saving the current settings whenever the application was exited. Thus, at the next application launch all settings would be the same as they were at the end of the previous session.

The major decisions made in coding this application are covered in the following subsections. Although timing loop, audio, video, haptic, and game scoring are considered separately for clarity, it is worthwhile to mention that all of these subsections are heavily interrelated. This often made troubleshooting difficult. For example, if the timing loop became erratic during testing, it was not immediately obvious as to whether this was a timing loop problem or if perhaps it was due to unexpected latency in the audio engine or latency in receiving commands from the GUI (Graphical User Interface). Additionally, differences between the simulator and an actual iPhone made it imperative to test on the device itself. The final design was the result of choosing the most straightforward approach to each of the following subsections that, when combined, yielded an application that would run on the actual device with stable timing.

## 5.1 Timing Loop (NSTimer versus NSThread)

A steady timing loop is central to the proper operation of this application. The central debate was whether to use NSTimer, NSThread with a while loop, or an NSTimer within an NSThread. Both the NSTimer and the NSThread with a while loop were attempted, with the more stable being the NSTimer version. It should be mentioned that initially the NSTimer version contained many NSLog statements to write to the console for troubleshooting. Writing to the console within a timing loop takes a short amount of time but the frequency of these write commands caused the timing to become erratic. Once the NSLog commands were commented out, the NSTimer version of the software produced a steady timing loop. Future work on the timing loop may investigate putting the NSTimer within an NSThread for even greater stability.

## 5.2 Audio (AVAudioPlayer versus OpenAL)

The role of audio in this application can be divided into two parts: the looped polyrhythm of snare and ride samples, and the samples that are triggered when the user taps either the LEFT or RIGHT button. To achieve this with the smallest impact on runtime processing power, all sound samples need to be loaded into buffers at startup. Next, consideration was given to which iPhone audio framework would be best suited to play the buffered audio. The simplest way to play an audio file on the iPhone is to use the AVAudioPlayer found in the AVFoundation framework. However, the documentation on AVAudioPlayer make its limitations clear:

"Apple recommends that you use this class for audio playback unless your application requires stereo positioning or precise synchronization, or you are playing audio captured from a network stream."

Since stereo positioning and precise synchronization were both of great importance, the OpenAL framework was chosen to handle sound. The documentation for OpenAL [7] points out that 32 note polyphony can be achieved and that sound sources need to be mono for the stereo positioning to work. A downside to using OpenAL is that it will only play certain file types and is very particular about the format. All of the Apple CAF (Core Audio Format) formatted files used for this project required re-formatting before they would play in OpenAL. The following Terminal command will properly reformat a file for use in OpenAL:

afconvert -f caff -d LEI16@44100 -c 1 in.wav out.caf

where in.wav can be an input file of any type and out.caf is the properly formatted output file.

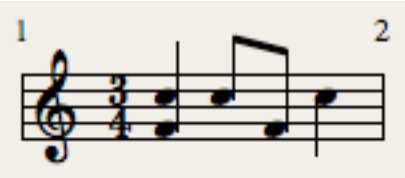
Initial testing of the timing loop that triggered the snare and ride samples revealed timing glitches. Further investigation revealed that the snare sample had a duration of 1 second and the ride cymbal had a duration of 8 seconds. Because of the long durations of these samples and the rapidity with which they were being triggered, the maximum polyphony

of OpenAL was soon exceeded, resulting in timing glitches. To remedy this, these samples were taken into an audio editor and reduced in duration to 500ms each, with a quick fade out. A retest of the timing loop with the new short duration samples revealed that the fix had resolved the timing glitches.

## 5.3 Visual (UIKit versus OpenGL)

Initially, very little consideration was given to using visual stimuli to convey polyrhythm. The only visuals in the first iteration of the design were two digital metronomes that each displayed the current measure and beat for each rhythm. The intention was to show that it is possible to use either the faster or slower rhythm as the basis for counting the polyrhythm. For example, the most common way to count a 4 against 1 polyrhythm is by treating the first rhythm like four quarter notes and treating the second rhythm as if it were a whole note. This would traditionally be counted as "1, 2, 3, 4" with the second rhythm being a whole note that occur on beat 1. An alternative way to count 4 against 1 would be to count the slower beat. In this method of counting, the second beat would be counted "1, 2, 3, 4" and the first beat would be counted as four 16$^{th}$ notes that occur every measure. Although the display of two digital metronomes conveyed this idea, it did so very awkwardly and after initial game testing this idea was abandoned. Should this idea ever be reinstated, it may be better illustrated by using metronomes similar to an analog clock.

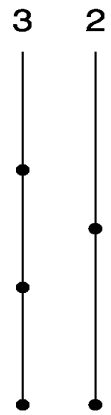
A second visual option that was given consideration was providing an actual written musical score for each polyrhythm. Figure 3a shows a 3 against 2 polyrhythm in traditional notation. Although this is feasible for a simple 3 against 2, anything more complicated soon becomes unwieldy to convey in a written score. Compounding the problem is that every combination of N against M would need a separate graphic. If N and M are each allowed to range from 1 to 16, this would mean creating 240 images. This approach was too cumbersome to be seriously considered.




(a) Traditional Notation

| | | | | | | |
|---|---|---|---|---|---|---|
| Left Hand | x | | x | | x | |
| Right Hand | x | | | x | | |

(b) TUBS (Time-Unit Boxes) Notation




(c) LSL (Line Segment Length) Notation

**Figure 3. A 3 against 2 polyrhythm illustrated in three different notations**

A third option was to use the TUBs (Time-Unit Boxes) notation as shown in Figure 3b. The TUBs notation is well suited to a computer application because it can be created programmatically. However, creating the grid for TUBs involves a considerable programming effort and also requires a large amount of horizontal screen space in order to be legible. Because of the screen size limitations of the iPhone, it would not be possible to use TUBs notation, despite its obvious advantages over traditional notation.

Out of necessity, the line segment length (LSL) notation shown in Figure 3c was developed. In this style of notation, dots represent the start of a note and the length of the line segments represent note durations. LSL is read from bottom to top, borrowing from the Guitar Hero paradigm. This notation can be thought of as vertically oriented TUBs notation without grid lines. It can also be thought of as similar to a piano roll. Two advantages of this notation become obvious. First, it is easy to see note start/stops and relative note lengths when two rhythms in LSL notation are placed next to one another. Second, to provide LSL style notation for every possible N against M polyrhythm would only require the creation of 16 static images (both N and M allowed to move between 1 and 16). Given the limited screen space of the iPhone and considering the relative ease of implementation, LSL notation was chosen for visual stimuli.

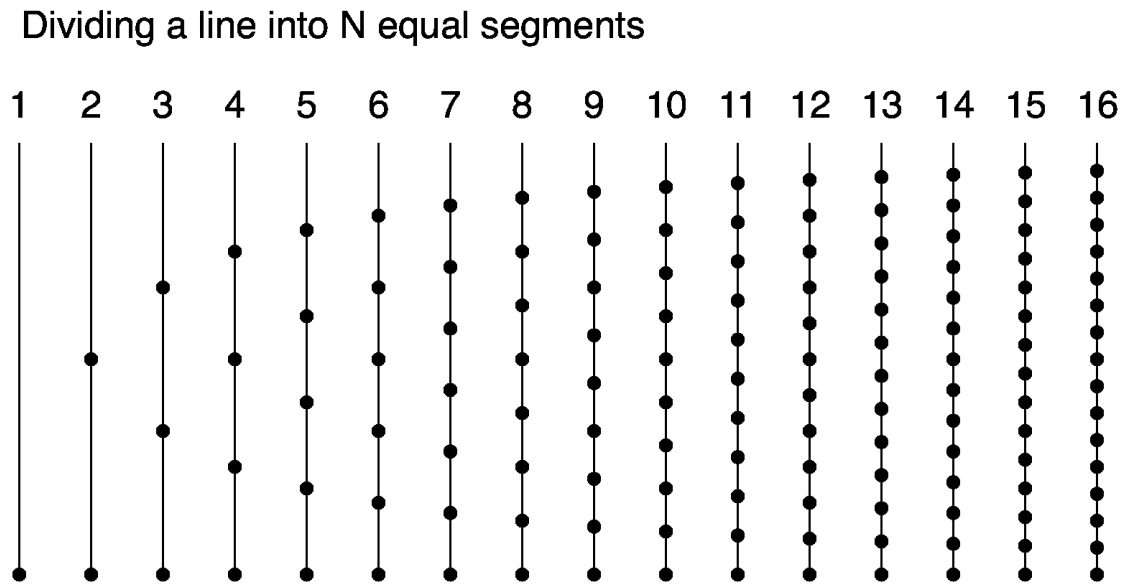



**Figure 4. Dividing a line into N equal segments**

Figure 4 shows a line divided into N equal segments, where N ranges from 1 to 16. These 16 images were all that was necessary for a complete LSL implementation.

## 5.4 Haptics (standard vibration versus jailbroken vibration)

The standard AudioToolbox framework allows for simple access to the iPhone's vibration feature. The actual call is:

```
AudioServicesPlaySystemSound (kSystemSoundID_Vibrate);
```

The simplicity of implementation comes at the cost of flexibility. Using the standard API (Application Programming Interface), it is not possible to change the vibration intensity or shorten its standard duration of 400ms. The only way to gain greater control over the vibration is to "jailbreak" the iPhone. For the first iteration of haptic design, a decision was made to work within the confines of the standard API. To use this one vibration effectively, the implementation would need to consider that vibrations should be spaced far enough apart that they would not overlap. Also, with only one vibration there could be no effective haptic distinction between the first and second rhythm. Given these considerations, the haptic vibration was assigned to the downbeat that both rhythms shared. This vibration would reinforce the first beat of a measure, which would always be a beat on which the user should press both the LEFT and RIGHT button simultaneously. Moreover, at a frequency of only once per measure, these vibrations would only be at risk of overlapping at extremely high tempos (when the length of a measure approaches only 400ms).

## 5.5 Game scoring (array with true beat times versus no array)

Very careful consideration was given to developing a fair and accurate game scoring algorithm. A diagram showing how taps are scored can be seen in Figure 5. The closer the user tap time is to the calculated time of the actual beat, the higher the score. User taps that fall within the "fine" region receive +6 points, those within the "finer" region receive +8 points, and those within the "finest" region receive +10 points. This method of scoring is similar to a game of darts, where the bull's eye has the highest point value and the points decrease in quantized levels as one moves further from the dartboard center. An accuracy of +/- 200 ms was required for the "fine" region, +/- 50 ms for the "finer" region, and +/- 10 ms for the "finest" region.

Rewarding accuracy is a step in the right direction but does not cover all scenarios fairly. A user should be penalized if they simply tap as often as possible in the hopes that a large number of the taps will score points. Also, completely missing a beat and not tapping at all should also result in penalty points. It was decided that both grossly inaccurate taps (ie. taps in the dead zone) and beats that go by without any tap should both be penalized with a score of -10 points. Figure 5 shows three tap pattern examples with their corresponding scores to help illustrate the scoring mechanism.

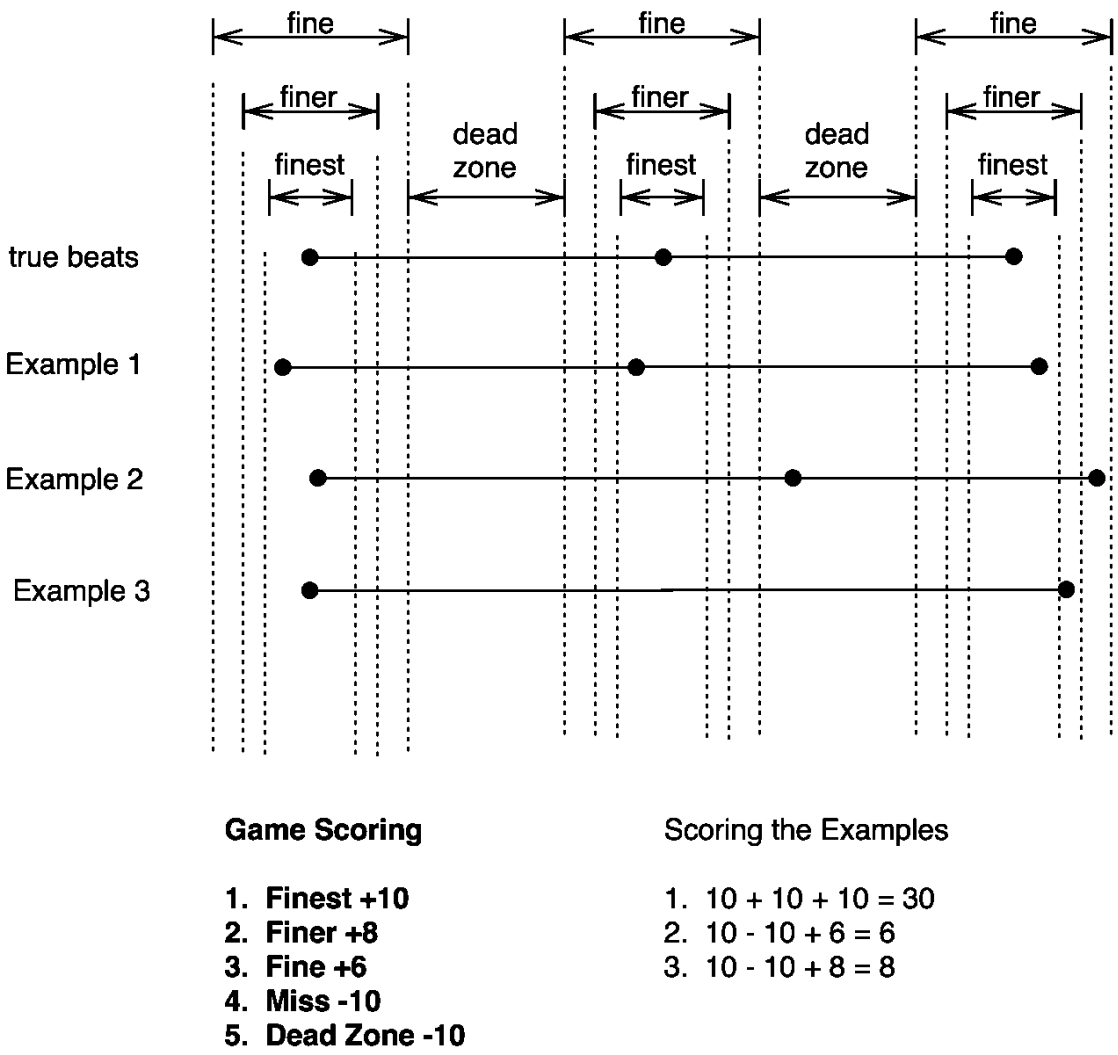



**Figure 5. Example Game Scoring**

The following explanation gives the specifics of the scoring algorithm. Before the start of a round, two arrays are filled: one containing the times of all the expected beats in the first rhythm and another containing the times of all of the expected beats in the second rhythm. These are declared as follows:

```
NSMutableArray *rhythm1TrueBeatTimes;
NSMutableArray *rhythm2TrueBeatTimes;
```

Anytime that the user taps the LEFT button the time of the button tap is compared to the times in rhythm1TrueBeatTimes and is scored accordingly. If the user tap time is even within the "fine" region, the time representing that downbeat is removed from the array. Anytime that the user taps the RIGHT button the time of the button tap is compared to

the times in rhythm2TrueBeatTimes and is scored accordingly. Again, if the user tap time is even within the "fine" region, the time representing that downbeat is removed from the array. The reason for removing items from these arrays is threefold. First, it reduces the size of the array that must be searched at every button press. Second, it prevents a user from scoring twice by tapping twice in rapid succession very close to the actual beat. Finally, any item left in one of these arrays at the end of a round represents a beat that the user let pass without tapping at all. The size of the rhythm1TrueBeatTimes array is calculated at the end of the round, multiplied by ten, and subtracted from the user's score for the first rhythm. Likewise, the size of the rhythm2TrueBeatTimes array is calculated at the end of the round, multiplied by ten, and subtracted from the user's score for the second rhythm.

## 6. Experiment

An experiment was conducted to determine if playing the game with all modalities of feedback turned on would improve a participant's ability to tap polyrhythms. The 12 participants that agreed to the study were tested on an individual basis in a soundproof studio in the basement of the SARC (Sonic Arts Research Centre) building. The participants were all SARC graduate students, 10 male and 2 female. None of the these individuals indicated an uncorrected visual, auditory, or motor impairment that might impact game play. There were 10 right-handed, 1 left-handed, and 1 ambidextrous participant. When asked to indicate a main instrument there were 3 guitar, 3 piano, 2 flute, 1 percussion, 1 oboe, and 1 bass guitar player. One participant indicated "none" as a main instrument.

Participants were asked to fill out a pre-experiment questionnaire in which they subjectively rated themselves from 0 to 10 in level of musical experience, sense of rhythm, video gaming experience, and finally either Guitar Hero or Rock Band experience. Across all participants, the mean and standard deviation for each of these categories is shown in Table 1.

**Table 1. Pre-Experiment Questionnaire Results**

| Category | Mean | Standard Deviation |
|---|---|---|
| Level of Musical Experience | 6.95 | 2.41 |
| Sense of Rhythm | 6.1 | 1.90 |
| Video Gaming Experience | 3.62 | 2.37 |
| Guitar Hero or Rock Band Experience | 0.47 | 2.47 |

To ensure consistency, every participant used the same iPhone 3G running OS 3.0 with Polyrhythm Hero version AT9 and audio running out of the built-in speaker at three-quarter of full volume. The settings shown in Table 2 were used for every participant.

**Table 2. Game Settings Used for Experiment**

| Parameter | Setting |
|---|---|
| Tempo: | 107 bpm in reference to the faster rhythm. |
| Count-in: | Recorded verbal count in that corresponds to the faster rhythm. In the case of a 7 against 4, the count in would be "1, 2, 3, 4, 5, ready, and". In the case of a 3 against 2, the count in would be "1, ready, and". |
| Spoken count | Off for both rhythms. |
| Tick-tock count | Off for both rhythms. |
| Audio | On for both rhythms (Rhythm 1 triggers a snare sampe, Rhythm 2 triggers a ride ride cymbal) |
| Static Visual | On for both rhythms. |
| Haptic Vibration | On, which means that a vibration occurs on the downbeat that both rhythms share. |
| Balance | Center panned for both rhythms. |
| Subdivisions | Varies with each example. |
| Measures Per Round | 2 |

Following an explanation of the game, participants were allowed to train on an 8 against 4, a 7 against 2, and a 2 against 7 polyrhythm for a combined total of 6 minutes. These particular combinations were chosen because they were not amongst the polyrhythms that appeared in the actual game. At the end of the training period, the participant was secluded in the studio room and left to play the game until either they had completed all ten polyrhythms or ten minutes had elapsed, whichever came first. The round score versus level scatter plot data for all twelve participants is shown in Appendix A, Figures A1 through A12. Analysis of this data shows that the majority of participants improved upon their baseline score for each polyrhythm with successive plays.

An anomaly in this trend, that can be seen clearly in the scatter plots, was the occasional baseline followed by one or more lower scores before the scores began to increase and eventually surpass the baseline. The post-experiment questionnaire revealed a possible explanation for this anomaly. Several participants claimed to first pay attention to the visual line segments and do their best to play the polyrhythm based primarily on visual cues. However, if they were unsuccessful, they would then focus primarily on the audio as a guide. The re-focusing of attention on a different modality could explain why some scores got worse before they got better.

One final piece of significant data on player improvement is that participant 5 spent 96 rounds attempting level 5 (a 3 against 2 polyrhythm). This participant was the only one of twelve not to improve on this level with successive plays. The fact that this participant listed "none" as a primary instrument may explain this finding. A larger participant pool would be beneficial to help identify the cause of these anomalies.

For users that completed all 10 levels in less than 10 minutes, the average number of attempts at each level is shown in Table 3. This data indicates that participants had the

most difficulty with the 4 against 5 polyrhythm, followed closely by 3 against 4 and then 3 against 5.

**Table 3. Average Number of Rounds Per Level for Participants Completing All 10 Levels**

| LEVEL | POLYRHYTHM | AVERAGE NUMBER OF ATTEMPTS |
|---|---|---|
| 1 | 1 against 4 | 1.5 |
| 2 | 2 against 4 | 1.625 |
| 3 | 6 against 2 | 1.25 |
| 4 | 3 against 6 | 1.125 |
| 5 | 3 against 2 | 2.5 |
| 6 | 3 against 4 | 6.375 |
| 7 | 3 against 5 | 5.625 |
| 8 | 5 against 3 | 2.75 |
| 9 | 4 against 5 | 6.5 |
| 10 | 7 against 4 | 3.75 |

After either level 10 was passed or 10 minutes had elapsed, the participant was asked to fill out a post-experiment questionnaire in which they were to subjectively rate statements from 0 to 10. The questions, along with mean and standard deviation across all participants are shown in Table 4.

**Table 4. Post-Experiment Questionnaire Results**

| Category | Mean | Standard Deviation |
|---|---|---|
| Game increased my understanding of polyrhythm | 7.81 | 3.34 |
| Game helped me to play polyrhythms better | 8.06 | 1.34 |
| Audio was helpful in playing the game | 7.45 | 1.48 |
| Video was helpful in playing the game | 7.16 | 2.32 |
| Haptic vibration was helpful in playing the game | 1.52 | 3.78 |

According to the post-experiment survey, the most useful modality of feedback was audio, with video a close second and haptic a distant third. To improve the haptic feedback, one user suggested a vibration of shorter duration. Perhaps haptics would be more useful for users that are depending on a downbeat in longer phrases or when the audio and video modalities are not present. It is interesting to note that the two participants that rated haptic feedback as extremely useful had the lowest scores. Perhaps paying attention to the vibration actually hurt scores.

Also according to the post-experiment survey, users were split as to whether animated visuals would make a better training tool. Proponents of animation argued that it would be easier to follow but opponents of animation argued that the current setup (static visual) makes the user rely more on the audio and is more closely tied to skills traditionally associated with musical training.

## 7. Future Work

There are several improvements on the game itself that are worth considering. First, experimental results suggest that the duration of the haptic vibration needs to be shortened in order to provide a more defined, impulse-like response. Currently, this is not possible with the standard API, but future releases of the iPhone SDK (Software Development Kit) may provide access to this functionality. Second, it would be very straightforward to add a difficulty setting (easy / medium / hard) that would simply reduce the size of the scoring bull's eye as the difficulty increased. This would give the user more control over the game and would be a simple way to add more challenge for advanced players. Third, consideration should be given to animating the static visual so that the line segment that represents the current beat is always shown in a different color. In the post-experiment survey, participants were split on whether animating the static visual would improve the game as a training tool. Consequently, if visual animation were to be added, a corresponding switch on the settings view would need to be added so that it could be turned on or off with ease. A final alteration worth considering is a version of the game that gradually removes modalities as the game progresses. The game might start off with animated visual, audio, and haptic cues. Gradually, the animated visual would turn static and then eventually disappear completely. The audio might get softer and softer and then be inaudible. By the end, users would be challenged to tap complex polyrhythms with only a single audio or haptic downbeat as a guide. This would certainly be a challenge for even advanced players and might prove to be a great training tool.

Beyond the preliminary study, numerous experiments can be run on even the current version of Polyrhythm Hero. An obvious follow up study is to experimentally determine which combinations of modalities are most effective in polyrhythm tap training. In the preliminary study, participants were asked to rate the relative effectiveness of the audio, visual, and haptic feedback. Because of the ease with which the individual modalities can be switched on or off, the relative effectiveness of each could be tested by increasing the participant pool and breaking them into the groups shown in Table 5.

**Table 5. Possible Test Groups**

| Group | Audio | Visual | Haptic |
|---|---|---|---|
| A | x | | |
| B | | x | |
| C | | | x |
| D | x | x | |
| E | x | | x |
| F | | x | x |
| G | x | x | x |

With sufficient participants in each group it would be possible to compare how, for example, Group A (audio only) improves over time as compared with Group D (audio and visual only).

Another worthwhile experiment might involve studying the long-term impact of game play. How would regular game play improve polyrhythm tapping over a period of days, weeks, or even months?

## 8. Conclusion

Initial experimental evidence and participant feedback suggests that this game is useful as a polyrhythm-training tool for instrumentalists. Based on participants' subjective modalities evaluation, the audio and visual cues were extremely helpful in game play. The haptic vibration did not fare as well, suggesting that the particulars of the haptic implementation should be revisited. Further testing with a larger participant pool could experimentally determine which modalities are most helpful in tapping a polyrhythm.

## References

[1] Deep Rhythm Lesson website. http://www.bobbrozman.com/tip_rhythm.html
[2] N. Gillian, S. O'Modhrain, and G. Essl. "Scratch-Off: A gesture based mobile music game with tactile feedback" in Proceedings of the International Conference on New Interfaces for Musical Expression (NIME09), Pittsburgh, USA, pp. 308-311, 2009.
[3] R. Malaka, J. Haeussler, and H. Aras. "SmartKom mobile: intelligent ubiquitous user interaction" in Proceedings of the 9th International Conference on Intelligent User interfaces, Madeira, Portugal, pp. 310-312. 2004.
[4] H. Anegg, G. Niklfeld, M. Pucher, R. Schatz, R. Simon, F. Wegscheider, T. Dangl, and M. Jank. "Multimodal Interfaces in Mobile Devices – The Mona Project" in Proceedings of the Workshop on Emerging Applications for Wireless and Mobile Access, NY. 2004.
[5] J. Hyndman, T. Lunney, and P. McKevitt. "AmbiLearn: Ambient Intelligent Multimodal Learning Environment for Children" in Proceedings of the 10th Annual PostGraduate Symposium On The Convergence of Telecommunications, Networking & Broadcasting, PG Net, Liverpool, pp. 277-282, 2009.
[6] G. Wang. "Designing Smule's Ocarina: The iPhone's Magic Flute" in Proceedings of the International Conference on New Interfaces for Musical Expression (NIME09), Pittsburgh, USA, pp. 303-307, 2009.
[7] OpenAL website. http://connect.creativelabs.com/openal/default.aspx
[8] Rhythm website. http://www.dolejsky.com/rhythm/
[9] "An Easy Method for Understanding and Playing Polyrhythms" by Bob Hinz. Mel Bay Publications, Inc. Creative Keyboard webzine. Feb 2008.

## Appendix A. Round Score Versus Level Scatter Plots

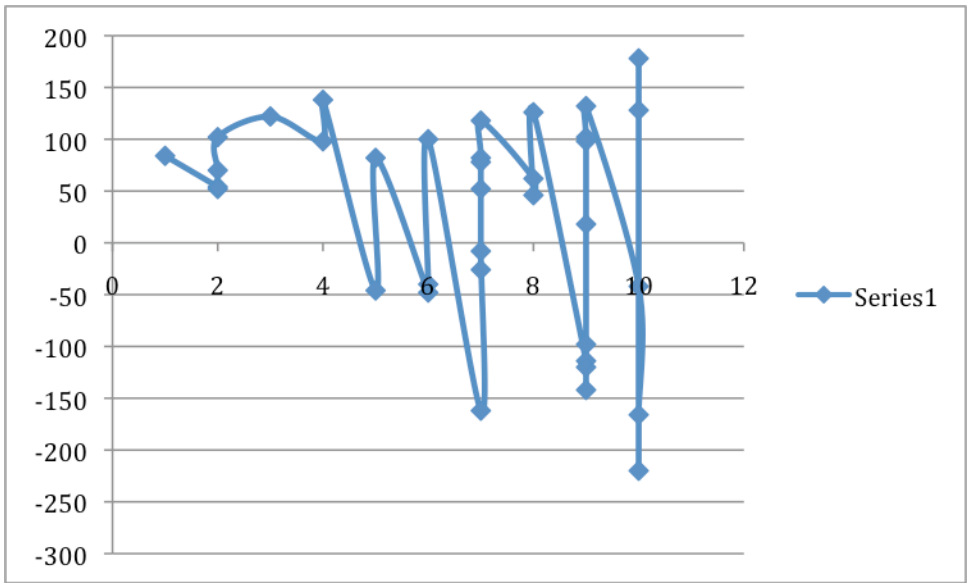



**Figure A1. Participant 1 Round Score Versus Level**

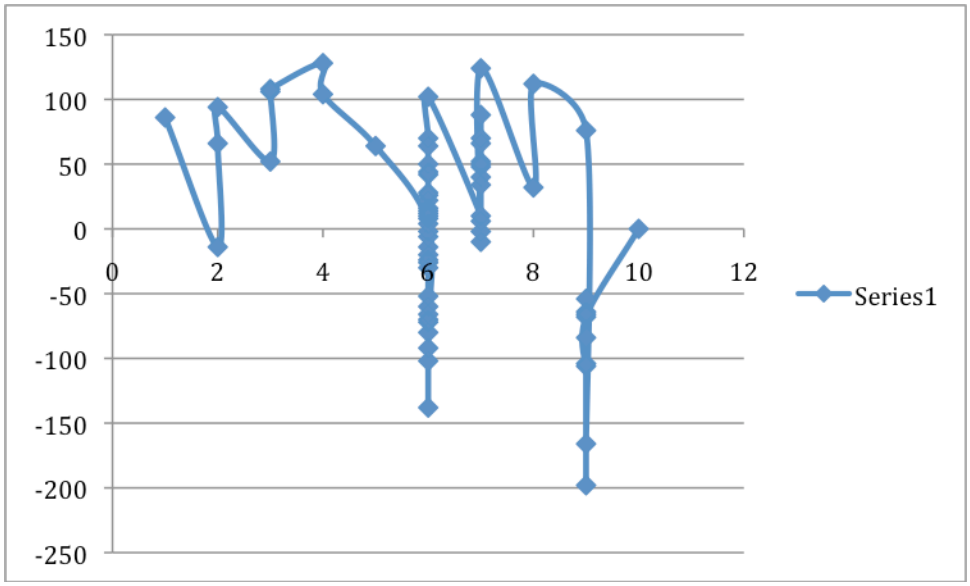



**Figure A2. Participant 2 Round Score Versus Level**

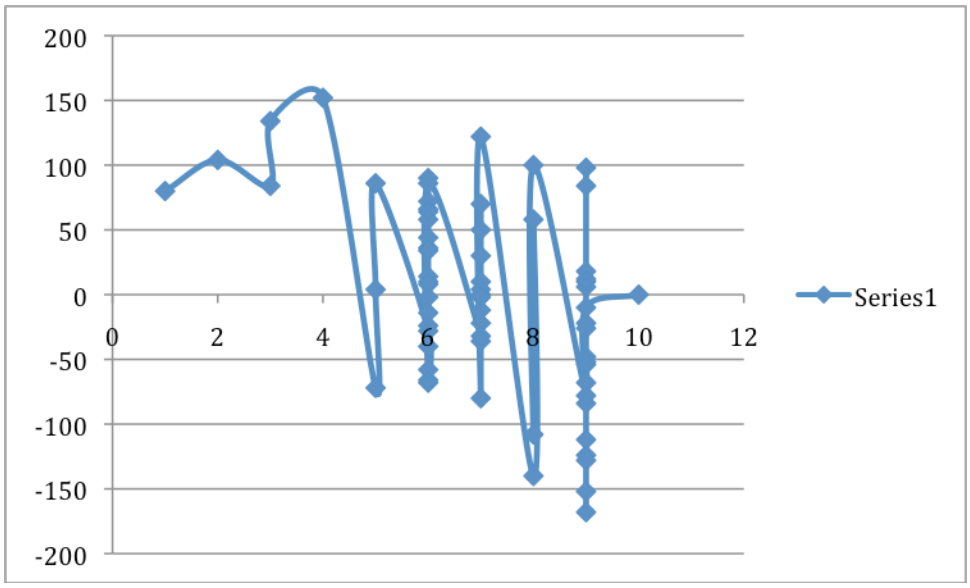



**Figure A3. Participant 3 Round Score Versus Level**

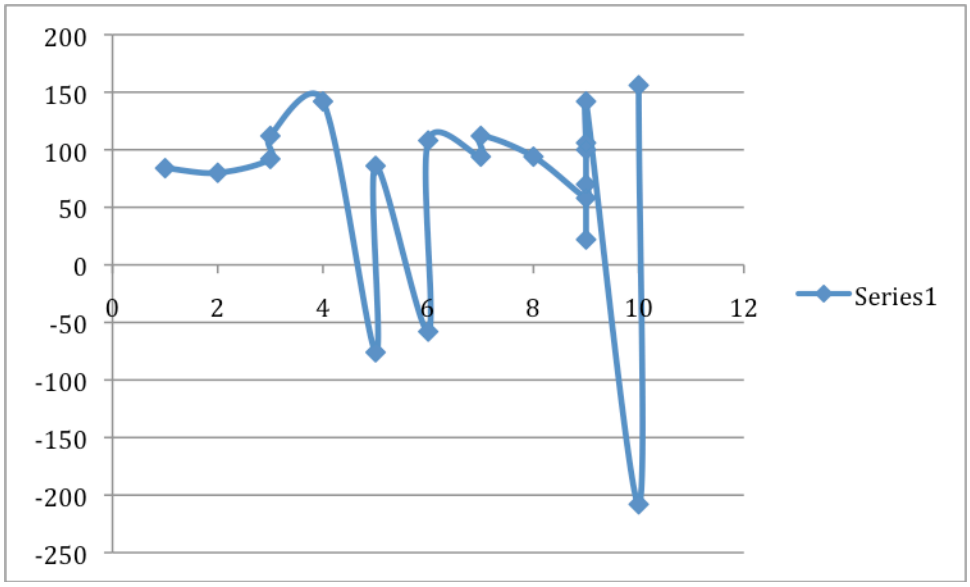



**Figure A4. Participant 4 Round Score Versus Level**

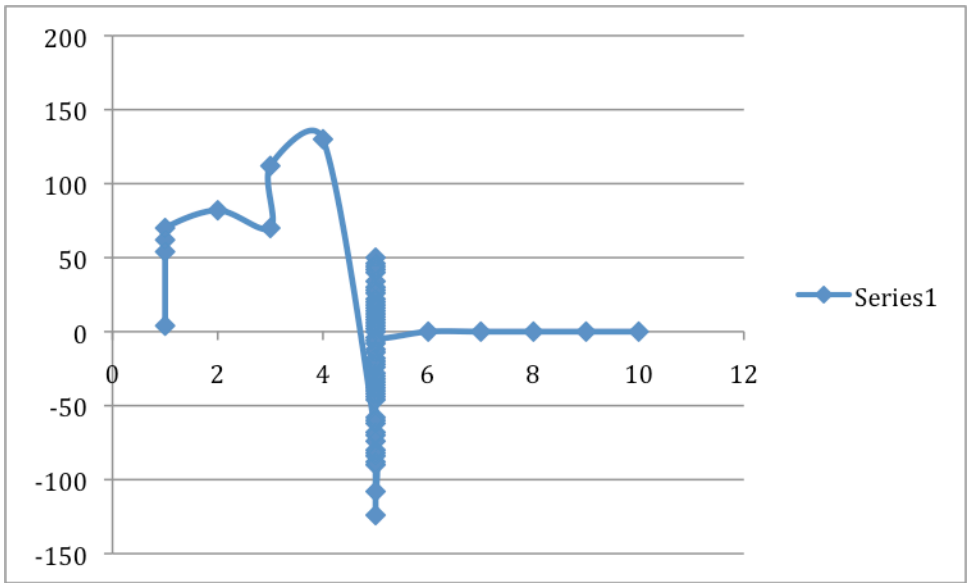



**Figure A5. Participant 5 Round Score Versus Level**

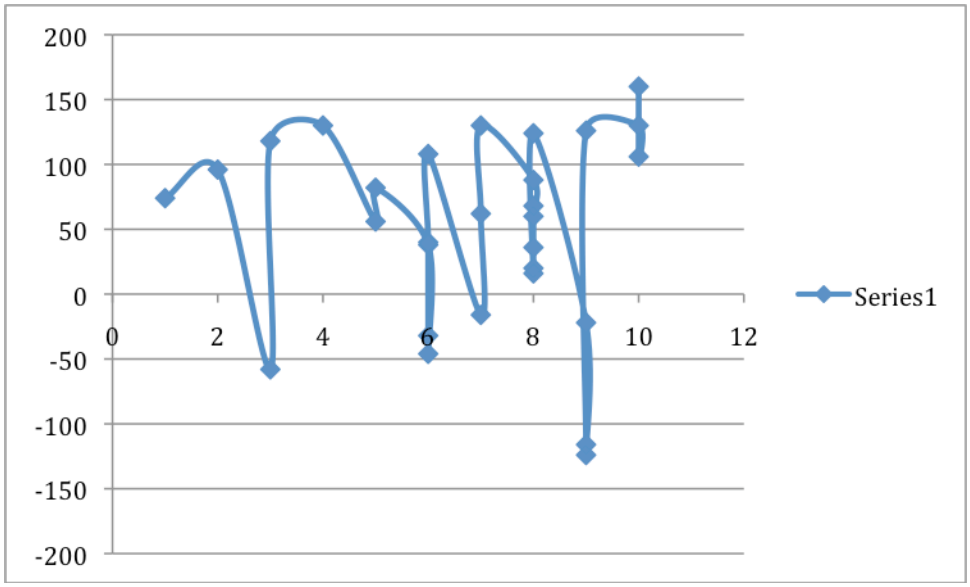



**Figure A6. Participant 6 Round Score Versus Level**

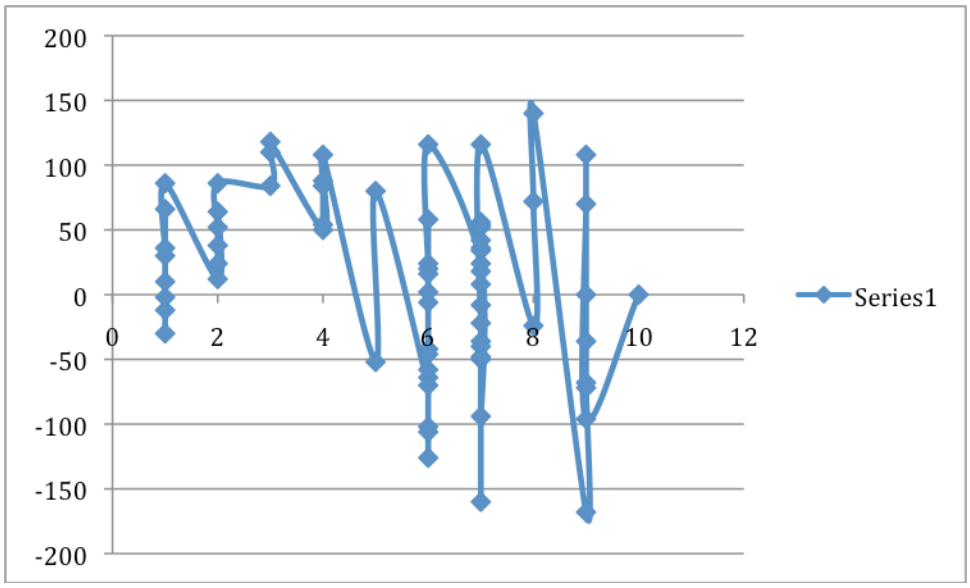



**Figure A7. Participant 7 Round Score Versus Level**

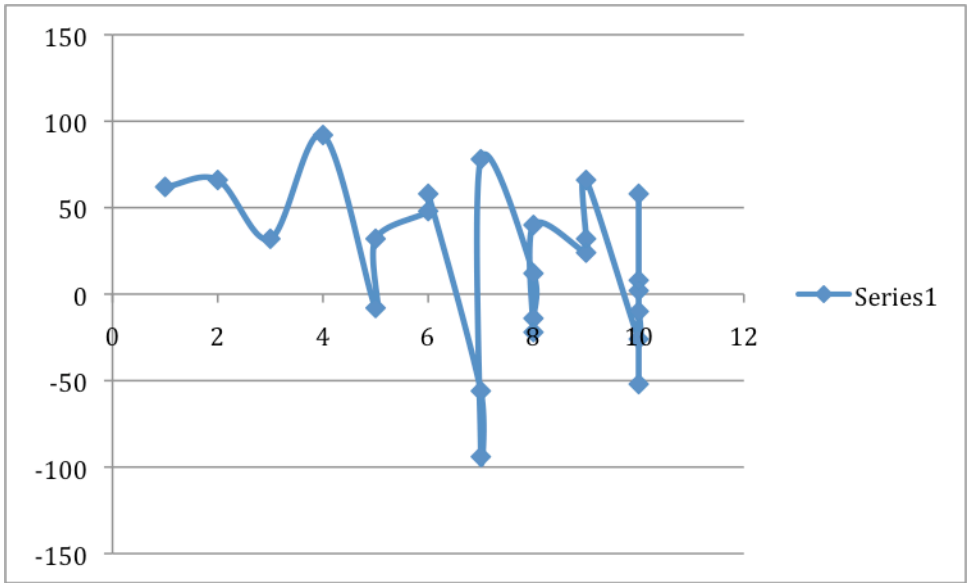



**Figure A8. Participant 8 Round Score Versus Level**

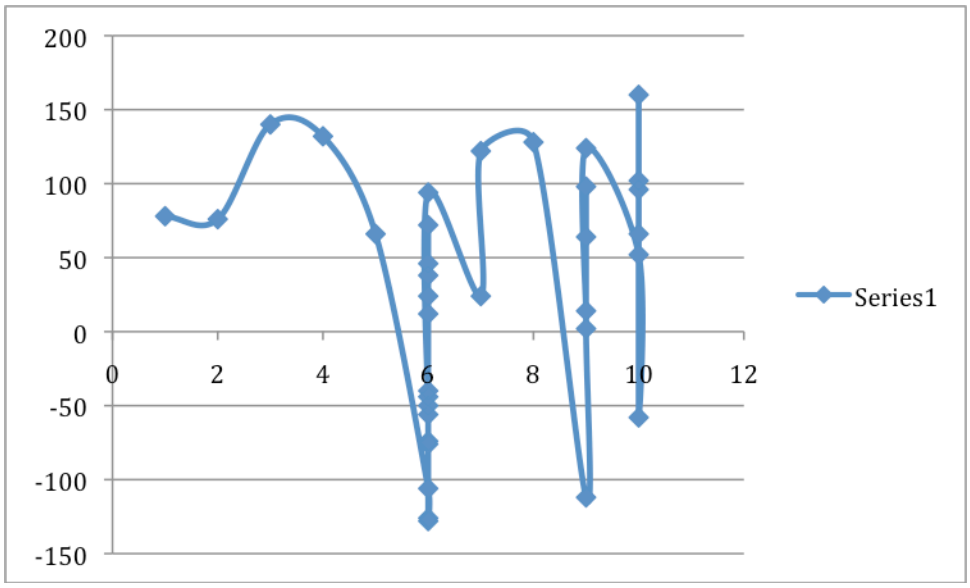



**Figure A9. Participant 9 Round Score Versus Level**

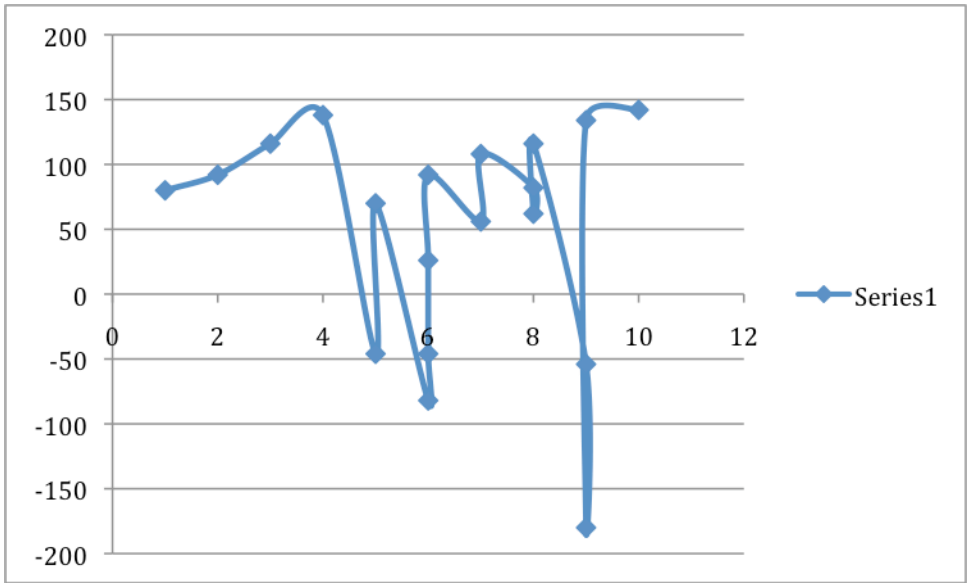



**Figure A10. Participant 10 Round Score Versus Level**

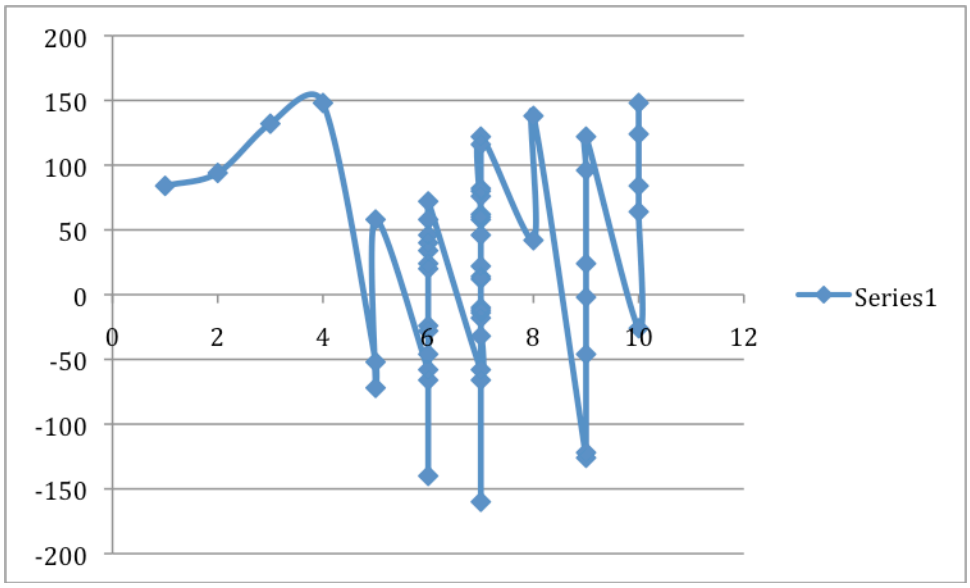



**Figure A11. Participant 11 Round Score Versus Level**

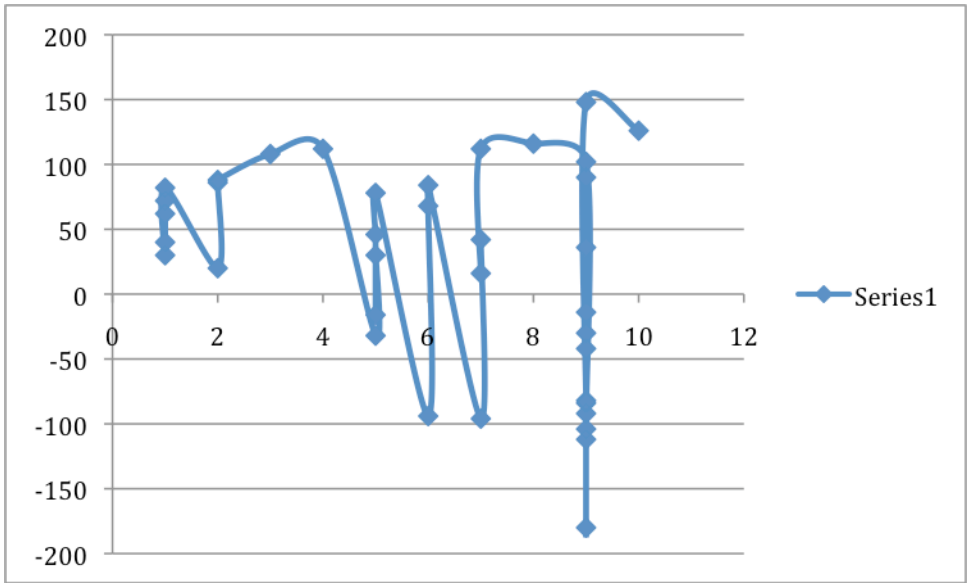



**Figure A12. Participant 12 Round Score Versus Level**