A Comparison of Accelerometer and Touch-based Input for Mobile Gaming

Stefania Medryk, I. Scott MacKenzie
Dept. of Computer Science and Engineering
York University
4700 Keele Street
Toronto, ON, M3J1P3 Canada
medryx@yorku.ca; mack@yorku.ca

Abstract - As gaming moves to mobile platforms, different input methods are incorporated to improve game design and gameplay. This is especially relevant with accelerometer-based tilt input which is supported on a variety of mobile games. We compared user performance during gameplay (level completion time, accuracy, score) and user enjoyment on *Bit.Trip Beat's* accelerometer and touch-based input systems. It was observed that while touch input yielded better results on gameplay statistics, tilt-based input was better received by the participants. Participants indicated that using the accelerometer made the game more challenging and more engaging. In addition, a substantial learning effect was observed for tilt-based input, while touch input demonstrated only a small learning effect.

***Keywords*:** Mobile gaming, accelerometer, touch input, tilt input, touchscreens, mobile computing

1. Introduction

With the increasing popularity and relevance of mobile devices, such as smartphones and tablets, there is greater interest in the variety of input methods possible. This is especially true for entertainment on mobile devices, as the market for mobile gaming currently dominates all other gaming platforms. As different control interfaces emerge, mobile game developers are considering different input options, including tilt as an input method.

However, in some situations, certain input options do not apply to or improve gameplay. Some gamers avoid tilt-based gaming, claiming it to be unnecessary, limiting, or too difficult in some situations. Yet to dismiss tilt input altogether is unwise as tilt can make gameplay more engaging if the game design is intended for, and compatible with, this input method.

In our study, we used an existing game called *Bit.Trip Beat* by Gaijin games (http://gaijingames.com/) to examine whether general gameplay results (level completion time, accuracy, score) differ between tilt-based input and touch input. In addition, we solicited user feedback on which input method is preferred for a game of this nature (arcade style mobile game).

1. 1. Related Work

Research on the use of tilt technology in human-computer interaction (HCI) extends back 20 years. Rahman et al. extensively studied the dexterity and limitations of tilt based input by examining the ranges of movement capable of being accessed through wrist movements using a mobile device (Rahman et al., 2009). Through the study it was discovered that using a single hand in controlling tilt, the movements of pronation and supination as well as flexion and extension should be the primary axes for tilt; quadratic mapping for discretization of tilt space also significantly improves performance across all tilt axes (Rahman et al., 2009). Applications of tilt on a mobile device include scrolling a document, menu navigation, changing screen orientation, zooming, panning, and so on (MacKenzie & Teather, 2012) .

Ruiz et al. (2011) conducted a study to allow users to define motion gestures to enter phone commands, thus eliminating the need to touch the screen or buttons. More recently, tilt-based input has also been applied to gaming. Yuan et al.'s work in *Tilt & Touch* displays the ability to access 6 degrees of freedom in mobile control using tilt a well as touch, and provides a template for using a mobile device as a 3D controller (Yuan et al., 2011). The addition of an accelerometer to gaming enables input control using device motion or tilt. This is easy to implement and requires no external calibration and minimal signal processing (Gilbertson et al., 2008). The cited study from 2008 described a no-button accelerometer-based game for a Nokia 5500 which was positively received by users. However the game may receive a different reception today, where smartphones and other "smart" prefixed devices allow for higher program complexity, especially in mobile gaming.

Accelerometers in mobile devices still have limitations. Full body movement detection and rotation cannot be achieved from the coordinates returned by the accelerometer (Whitehead et al., 2011). This puts a limitation on how to successfully design a game using the accelerometer as an input method. Gilleade et al. (2004) point out that when there is a failure to correctly operate the game device or interface, "at-game" frustration occurs whereby the interface itself is the focus of attention. In a sense this is the opposite of "transparency," whereby the user fully engaged in gameplay and is barely aware of the interface. Clearly, considerable scrutiny is warranted in choosing an input method for mobile games. Our research is intended to provide an examination of tilt vs. touch in mobile gaming.

1. 2. Bit.Trip Beat

For our experiment, we used a pre-existing and rather simple game called *Bit.Trip Beat*. Gameplay uses either touch or tilt input. It is a retro-themed game reminiscent of *Pong*. See Fig. 1. The user manoeuvers a paddle vertically (by tilting the device forward or backward or by touching the screen) to bounce back incoming pixels. Without misses, forty pixels are launched for the first level level; these change depending on misses, and more pixels are launched if the user misses pixels. The game has three main levels, each with eight stages. The user progresses through each stage by successfully hitting pixels with the paddle and bouncing them back. With each hit, a green bar increases in size and the user gains points. Once the green bar is entirely filled, the user progresses to the next stage. If the user misses a pixel, a grey bar expands to tally losses. Gameplay includes audio feedback: hitting pixels adds to a melody while missing pixels makes a "thunk" sound. There is no vibrotactile feedback.

Fig. 1. Screenshot of *Bit.Trip Beat* in level 1-1.

If the grey bar fills up entirely, game-over occurs. Also, when hitting many pixels consecutively, a *combo score multiplier* is invoked and the green bar fills faster. Therefore, the game works such that hitting pixels consecutively without misses allows for a faster progression through levels, while less

accurate play results in a slower progression through levels. The combo score multiplier is shown at the bottom left in Fig. 1.

2. Method

In order to investigate differences between the touch and tilt input for the same game, a user study was conducted where participants played a game using both input methods. We were also interested in comparing the learning patterns for the two input methods. For this, we used the power law of learning. This section describes the methodology used, including the independent and dependent variables and how the data were acquired.

2.1. Participants

Eighteen participants were recruited for the experiment. Participants were approached in person at the local university campus and asked to partake in the experiment. All participants were students in the digital media or computer science programs. Ages ranged from 18 to 31 years (*mean* = 22.4, *SD* = 2.8). Of the 18 participants, 12 were male, 5 were female. The consensus with all participants was that they had some experience with gaming either on a mobile platform or on other devices.

2.2. Apparatus

A smartphone device was used as the test apparatus. A case was added to prevent damage to the device and to mimic typical use (most users have a case for their device when using it and while in transit). The device was a Samsung *Galaxy SII LTE* running Google's *Android Jelly Bean 4.1.2*. Dimensions for the device were 13.1 cm × 6.99 cm, and with the case attached 13.5 cm × 7.50 cm. See Fig. 2.

Fig. 2. The test device was a Samsung *Galaxy SII LTE*.

For the experiment, we used Transition, the first stage of the first main level of *Bit.Trip Beat*. This is the simplest stage and is sufficient for participants to get accustomed to the gameplay and to the input method. Therefore, it was able to fulfil the purpose of our user study.

In addition to the device and game, a short questionnaire was used. Five items were presented before testing, and one item was presented after testing. The initial five items asked about the participant's age, gender, initials, experience with mobile gaming, and platform most often used for mobile gaming. The last item asked the participant to indicate which input method they found most enjoyable.

2.3. Procedure

Participants were briefed on the goals of the research following which they were seated and given the first portion of the questionnaire. Once completed, each participant was given 30 seconds to practice the initial input method, and then testing began. After 5 trials for the first input method, participants were

given another 30 seconds to become accustomed to the second input method, and were tested for 5 trials again. Fig. 3 shows a participant performing trials using touch input.

The data were gathered manually, with the experiment conducted in an observational manner. Performance differs in level completion time, misses, and score. During each trial, the user was manually timed on level completion; total misses were counted manually and the final score for each level was noted and recorded on a log sheet. After the experiment the data were transcribed into a spread sheet for follow-up analyses.

Participants were directed to complete only the first stage of the level; once the green bar at the top filled, the screen shifted to the next level. At this point participants were directed to stop (press pause) where the time, score, and total number of misses were recorded. If the level terminated due to the grey miss bar fully expanding, game over was declared and a score of 0 was recorded.

Fig. 3. A participant doing the experiment using touch input.

Once testing was complete, an additional question was presented to the participant on which layout they enjoyed more and why.

2.4. Design

The study was a 2×5 within-subjects design. There were two independent variables:

- Input method (tilt, touch)
- Trials (1, 2, 3, 4, 5)

The dependent variables were level completion time, error rate (number of misses), and final score per level. In addition, data were collected via the questionnaire administered before and after testing.

Participants were divided into two groups to counterbalance the order of testing and thereby cancel learning effects due to the sequence of testing. One group was tested on tilt, then touch, while the other group was tested on touch, then tilt. Each participant completed 5 trials of level 1-1 of the Transition stage in the game, using both tilt and touch input. Testing lasted approximately 15 minutes per participant. The total number of trials was 18 participants $\times$ 2 input methods $\times$ 5 trials = 180 trials.

3. Results and Discussion

The data from the questionnaire and experiment were transcribed into a spread sheet where summary measures were calculated and charts were created. Analysis of variance results were performed using Anova2, a free Java-based Internet download (http://www.yorku.ca/mack/HCIbook/).

3.1. Initial Questionnaire

From the results of the initial questionnaire, it was found that all users had some experience with mobile gaming. On frequency of mobile gaming, 5 participants indicated "often," 7 indicated "sometimes," 6 indicated "rarely." No user responded "never" to the question. Of all the participants, 8

use an android smartphone as their main device for mobile gaming, 5 use an iPhone, 3 use an iPad, while 2 use a Blackberry.

The performance results are given next, organized by dependent variable. Note that the group effect for counterbalancing was not statistically significant for all three dependent variables ($p > .05$). Thus, counterbalancing had the desired effect of cancelling any learning effect due to the order of testing.

3.2. Level Completion Time

Trials with a perfect score and no misses had a shorter level completion time; levels with more misses generally took longer. In addition, if a participant missed many pixels in a row and quickly filled the grey bar for misses, game-over occurred. This generally took less time than if the user missed a few pixels yet completed the level. The minimum recorded level completion time was 50.0 s with 0 misses, while the highest was 102.6 s with 8 misses.

The grand mean for level completion time was 56.1 s. With tilt-based input, the mean was 59.5 s and with touch input the mean was 52.6 s. Thus, touch input was 13.2% faster than tilt input. See Fig. 4a. The effect of input method on level completion time was statistically significant ($F_{1,16} = 19.5, p < .0005$).
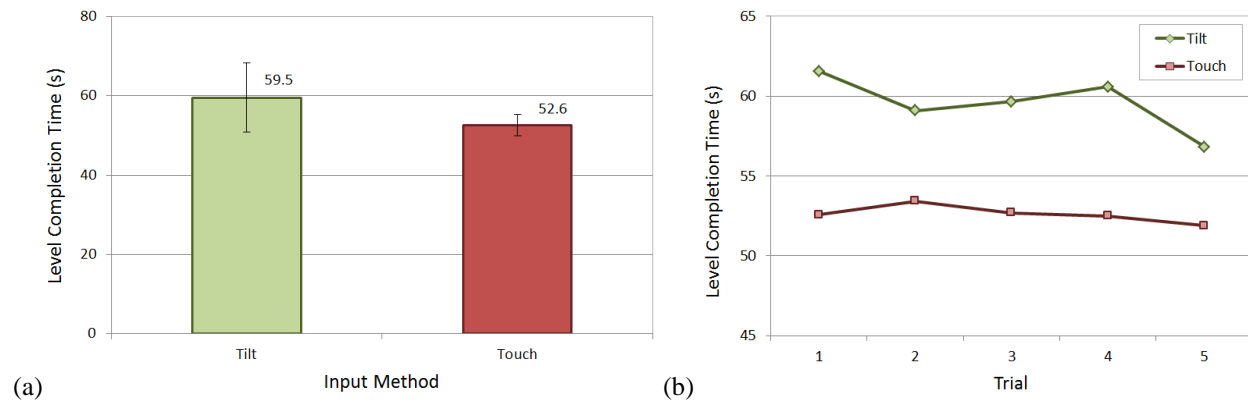

Fig. 4. Level completion time (a) by input method and (b) by trial. Error bars show ± 1 *SD*.

Level completion times were recorded for each trial. A modest improvement with practice was observed, as seen in Fig. 4b. (Each point in Fig. 4b is the mean of the observations for 18 participants.) Overall, the decrease in level completion time was 4.7%, from 57.1 s for the first trial to 54.4 s for the last trial. As the decrease was small, the effect of trial on level completion time was not statistically significant ($F_{4,64} = 1.26, p > .05$). Even still, it is evident in the figure that the improvement with practice was greater for tilt-based input than for touch input (6.2% vs. 1.1%). This follows from the simple fact that touch input was familiar to all participants and thus afforded less opportunity for learning.

3.3. Accuracy

With regards to accuracy, the lower the miss count the more accurate the user is with the control system. Missing 0 pixels results in a perfect score for the level. The highest observed miss count for this experiment was 25 while the lowest was 0. The grand mean for accuracy was 3.65 misses per level. By input method, the means were 5.98 for tilt and 1.32 for touch. See Fig. 5a.

As seen in Fig. 5a, touch input is 352% more accurate than tilt-based input. This is due to participants' familiarity with touch input, as well as to the fact that touch input is an example of *direct input* (the finger is the pointer and directly controls where the paddle will go, simplifying the task). Despite considerable variability by participant (particularly for tilt-based input; see error bars in Fig. 5a), the effect of input method on accuracy was statistically significant ($F_{1,16} = 21.9, p < .0005$). Clearly, there were some interaction issues with tilt-based input, however. Of the 18 participants, two averaged less

than one miss per 40 pixels in a trial, while three averaged more than 10 misses per 40 pixels in a trial. Evidently, tilt-based input will pose a challenge for some users.
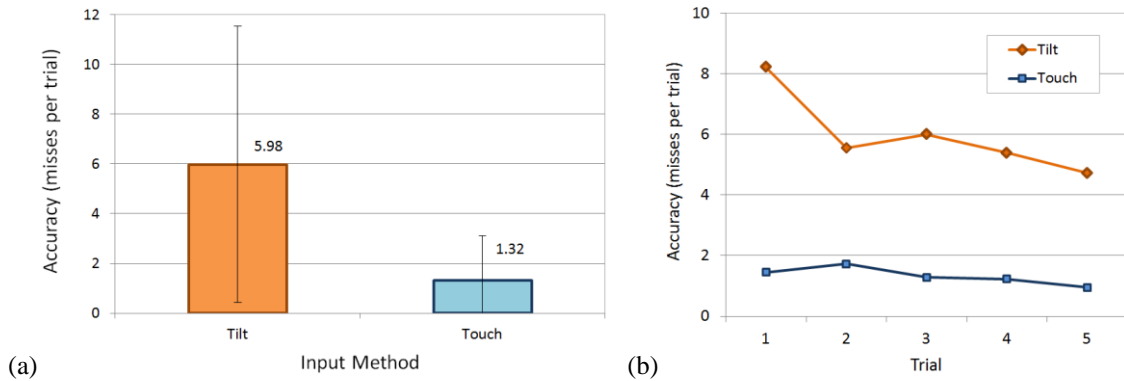

Fig. 5. Accuracy (a) by input method and (b) by trial. Error bars show ±1 *SD*.

Learning progress was also observed over the trials. Fig. 5b displays the learning progression on accuracy between the two input methods over the 5 trials. As observed from the 1st to 5th trial, there is a stark drop in the number of misses using tilt-based input (22.7%), while with touch input accuracy is generally quite good with the number of misses decreasing less so over time (12.4%). Despite the improvement for tilt-based input, it does not reach the level of accuracy that touch input allows.

3.4. Score

Some of the data depicting the score is contradictory with the accuracy and level completion time. This is on account of the combo score multiplier which occurs when individuals score multiple successful hits in a row. This increases the score on each hit and skews the score data. When an individual stays in the game longer, even if they miss multiple times, they often benefit from a combo score multiplier. Thus, they may obtain a higher score than an individual that missed much less. Despite the variability this causes, there was a statistically significant effect of input method on score ($F_{1,15} = 37.4, p < .0001$).

The highest possible score an individual can get on a level is 2775. This score implies that they played a perfect level with no misses. The grand mean for score was 2378, while the mean score by input method was 2548 for touch input and 2208 for tilt-based input. Thus, touch input yielded a 15.4% better score on average than tilt-based input. See Fig. 6a.
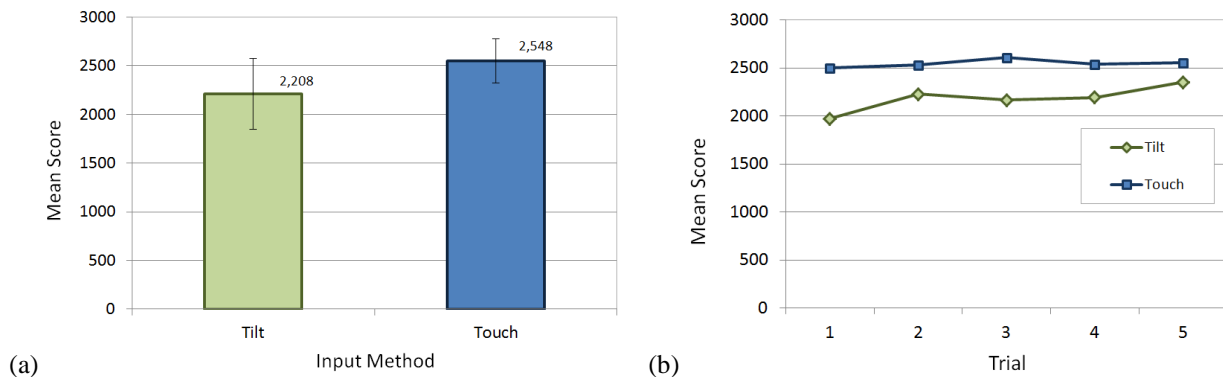

Fig. 6. Mean score (a) by input method and (b) by trial. Error bars show ±1 *SD*.

As noted above, if game-over occurs without bouncing back any pixels, a score of 0 is recorded. In all, 4 instances of this were observed, all with tilt-based input.

As with the previous observations on accuracy and level completion time, score also improved over time, more so with tilt-based input. This is seen in Fig. 6b. As the improvement was modest overall, it follows that the effect of trial on score was not statistically significant ($F_{4,64} = 1.85$, $p > .05$).

3. 5. Power Law of Learning

For each of the three dependent variables, we built models using the power law of learning, with an extrapolation to the 10th trial. Overall, the models provided only modest explanations of the learning effect. This was likely due to the limited amount of practice in the experiment – only 5 trials. The models for level completion time are shown in Fig. 7. As expected, the model for tilt-based input is a better predictor ($R^2 = .471$) than the model for touch input ($R^2 = .239$). This occurs because there was less room for improvement with touch input, due to participants' prior experience with touch-based control, as noted above.
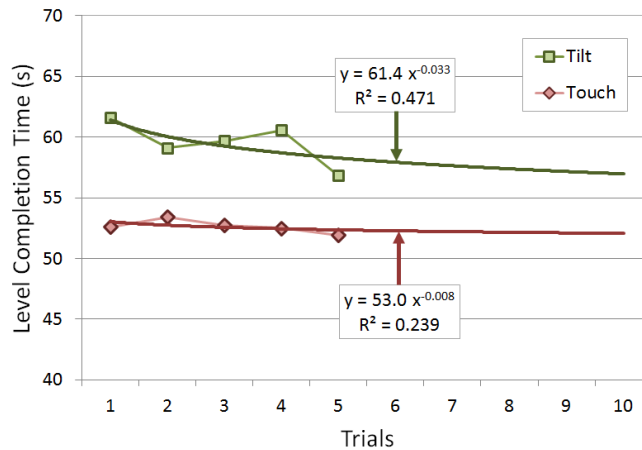

The equations shown in the figure:

$$y = 61.4\, x^{-0.033}$$
$$R^2 = 0.471$$

$$y = 53.0\, x^{-0.008}$$
$$R^2 = 0.239$$

Fig. 7. Power laws of learning for level time completion time with extrapolation to the 10th trial.

3. 6. Participant Feedback

With regards to the post-experiment questionnaire item, 13 (72%) of the participants indicated a preference for tilt-based input over touch input. The main reason cited was that it was more challenging and more engaging to use the accelerometer and tilt the device to control the paddle and bounce back pixels. Some participants indicated that touch input was too easy for this type of game. However, the 5 (28%) participants who preferred touch input cited reasons of either simplicity or gratification. A few participants found the accelerometer control too difficult, while a few participants preferred the familiarity and the success that touch input offered.

4. Future Work

As the power law of learning was reported, future work should include more extensive testing with the two input methods, in addition to exhaustively testing the same level and extra stages. As well, different games of the same style (arcade) should be tested in order to sufficiently elicit opportunities where tilt-based input is appropriate and appealing. In addition, other devices such as tablet PCs should also be tested to understand users' levels of comfort and engagement with tilt-based input using devices of different sizes. Finally, a wider range of participants (different age groups and backgrounds) should be tested in order to have a broader understanding of consumer demand and opportunity.

5. Conclusion

In this study, better gameplay results were recorded using touch-based input based on level completion time, accuracy, and score. However, little improvement was observed for touch-based input over the course of five trials of testing. Tilt-based input, on the other hand, showed greater improvement

with practice. Despite not performing as well with tilt-based input, the majority of participants chose tilt as their preferred input method, citing tilt as being more enjoyable and more engaging. We conclude that this control set functions well for this game and deserves credit for making the gameplay appealing for users. However, these results may not apply in other situations for mobile gaming. *Bit.Trip Beat* succeeded in using tilt-based input to enhance gameplay due to its simplicity. The classic retro feeling of the *Pong*-like game was enhanced and given a fresh new take by using tilt-based input.

References

Gilbertson P, Coulton P, Chehimi F. (2008). Using "tilt" as an interface to control "no-button" 3-D mobile games. Computers in Entertainment 6, no 3: 38.

Gilleade K.M., Dix A. (2004). Using frustration in the design of adaptive videogames. "Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology," ACM pp. 228-232.

MacKenzie I.S., Teather R.J. (2012). FittsTilt: the application of Fitts' law to tilt-based interaction. "Proceedings of the 7th Nordic Conference on Human-Computer Interaction – NordiCHI 2012," ACM, pp 568.577.

Rahman M., Gustafson S., Pourang I., Subramanian S. (2009). Tilt Techniques: Investigating the Dexterity of Wrist-based Input. "CHI '09 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems," ACM, pp 1943-1952.

Ruiz J., Yang L., Lank E. (2011). User-defined motion gestures for mobile interaction. "Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI 2012," ACM, pp. 197-206.

Whitehead A., Johnston H., Fox K., Crampton N., Tuen J. (2011). Homogenous accelerometer-based sensor networks for game interaction. Computers in Entertainment, 9, no. 1:1.

Yuan D., Haoyi R., Gang P., Shijian L. (2011). Tilt & Touch: mobile phone for 3D interaction. "UbiComp '11 Proceedings of the 13th international conference on Ubiquitous Computing," ACM, pp 485-486.