Tilt-Controlled Mobile Games: Velocity-control vs. Position-control

Catalina I. Constantin & I. Scott MacKenzie
Dept. of Electrical Engineering and Computer Science
York University
Toronto Ontario Canada M3J 1P3
catalinaioanaconstantin@gmail.com & mack@cse.yorku.ca

***Abstract* — We examined tilt-control for mobile gaming. An application was created using device tilt to move a virtual ball through a simple maze. The tilt of the device was mapped to ball movement in two ways: velocity-control and position-control. A user study with 12 participants compared the orders of control for performance (speed and accuracy), ease of use, and user preference. Results indicate that movement using position-control is about 16% faster than movement using velocity-control. On measures of accuracy, there was no significant difference between the two orders of control. Qualitative measures indicated no significant difference between the two orders of control on ease of use, although 10 of 12 participants indicated a preference for velocity-control (despite the better performance with position-control).**

Keywords—mobile games; accelerometer-based tilt-control; order of control; velocity-control; position-control

I. Introduction

In recent years, the use of mobile devices has greatly increased. Most such devices include embedded components to enhance the user experience. Cameras, light sensors, vibro-tactile actuators, accelerometers, gyroscopes, GPS receivers, and so on, work together in broadening the experience for users. With the prefix "smart", these devices provide unprecedented connectivity to users, through both 3G and Internet access. Applications span a wide range, from social networking to business needs to entertainment. Entertainment applications, otherwise known as games, come in many forms and with many themes. The focus in this paper is entertainment applications that use the device's built-in sensors as input for game-play.

One common sensor on mobile devices is an accelerometer, which provides data on the angular movement of the device (pitch, roll, azimuth). Accelerometer data are useful for many tasks, such as document scrolling [1, 6, 13], panning and zooming [16], pointer control [2, 10], menu navigation [15], and changing screen orientation [8]. A recent addition is games [5, 11]. Using data from the accelerometer, device tilt becomes a method of input control during game-play. In fact, the use of tilt input for games is now commonplace; however, tilt is generally used to control the velocity of an object or scene view. Many games currently exist, such as *Fish Frenzy* and *Glow Maze*, which map device tilt to object velocity; while very few map device tilt to object position.

With tilt already in use, it is important for mobile game developers to consider which control method is better or preferred for the game and context of use. For this paper, *order of control* refers to the mapping of device tilt to a controlled object in a mobile game. The possibilities are *position-control* and *velocity-control*. With position-control, the angular position of the device (relative to a home position) has a direct correspondence to the position of an object. With velocity-control, the angular position of the device has a direct correspondence to the movement (velocity or acceleration) of the object.

The choice of one order of control over another might impact a game's difficulty or enjoyment. Thus, our goal is to empirically investigate if one order of control is easier to use, and also if order of control bears on user preferences. We address several questions: What effect does order of control have on the overall speed and accuracy of game-play? Is the difference slight or is it significant and relevant for mobile game design? Finally, do users prefer one order of control to the other, or is the difference minor? The answers to these questions may depend not only on whether position-control or velocity-control is used, but also on context of use, or type of task. We will revisit this point shortly.

In the next section, previous studies on tilt-control and its applications are reviewed and examined. Then, the methodology of our user study is discussed, followed by a presentation and discussion of the results. Finally, conclusions are drawn about which order of control performed better in terms of quantitative results and user preferences.

A. Related Work

Previous studies show that the input method for games is important in the development of games [5]. However, the use of tilt is relatively new, since game development has only recently expanded to mobile devices. Console and PC games have different form factors to consider when developing a game. With a PC, for example, a player does not lift the keyboard and tilt it during game-play. However, that is not to say that tilt-control only exists on mobile devices. Gilbertson et al. note that the Nintendo *Wii Remote* controls game-play using a built-in accelerometer in the hand-held controller [5]. They cite an interview with Nintendo that noted "when playing a game, the nearest thing to the player is the controller. The controllers should therefore be regarded as an extension of the player rather than as part of the console." [5, p. 38:2]. This brings forth the point that using tilt as a method of control can be seen as a natural extension of the player. For example, the tilting motions used on a physical marble-maze game (see Fig. 1) may transfer to a virtual version on a mobile device.

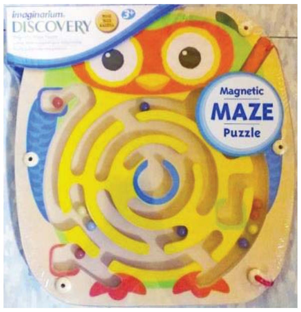

Fig. 1. Wooden marble-maze game.

Medryk and MacKenzie tested game-play with a Pong-like game (*Bit.Trip Beat*) using touch-input and tilt-input [11]. Each input method was used to control the position of a paddle to strike in-coming balls. They observed that while touch input yielded better quantitative results, tilt input was preferred by participants. Rather than shunning tilt input because of control difficulties, users reacted positively, saying that the game was more challenging and more engaging using tilt-input. The research also reported a substantial learning effect for tilt-based input, while touch input demonstrated only a small learning effect. This was likely due to participants' experience with touch input for common interactions on their smart devices.

The same research noted that some gamers may avoid tilt-input since they find it more difficult for game-play [11]. This seems contrary to the point above that device tilt is a natural method of control. However, the same can be said of a physical tilt-based game such as the aforementioned marble-maze. In short, the interaction is natural *and* difficult. Conversely, it was also noted that "tilt can make gameplay more engaging if the game design is intended for, and compatible with, this input method" [11, p. 117-1].

Fitton et al. tested tilt-based input in a target selection task under different conditions of mobility (sitting, walking) and grip (one-handed, two-handed) [4]. They also observed a learning effect but noted that while tilt-control is not necessarily the simplest input method users won't reject it due to its difficulty. In fact, Fitton et al. also reported that no participants had difficulty learning the technique and that the use of tilt input is promising [4].

While an engaging game is what users seek, Bucolo et al. remind us that game developers look for ways to "take advantage of the unique functionality provided by the mobile phone" [3, p. 87]. Their research showed that mobile devices are easy to manoeuvre and this translates into using tilt-input for a variety of interactions, including panning [3]. Panning is the use of a device's camera or position sensor to track a user's movements and convert them to the movements of a controlled view in the game. These physical manipulations lend themselves to making an application respond more naturally to the way the device is handled.

Apart from the concept that desktop computers and gaming consoles may not afford the same methods of input as mobile devices, the reverse also holds true. Rekimoto maintains that the most important difference between a mobile device and a desktop computer or a console is the allotted screen space [15]. This highly influences the decisions of game developers in many aspects of game design, including method of input. The user interface on a desktop computer is largely impractical if scaled to a mobile device. There are many controls that would be too small to see, and other elements that would occupy too much screen space on a small mobile device. Even touch-input might not be practical for some games since touching the display surface may occlude the game scene. Rekimoto also posits that using other input methods, such as tilt-control, allows for single-hand operation [15], which can be important at times, such as when a user is multitasking.

Concerning the context of use, or type of task, we turn now to work by Teather and MacKenzie [17]. On the surface, their research seems similar to ours. They compared tilt-based position-control to tilt-based velocity-control and found that position-control was superior. Position-control tilt-input yielded better performance and was preferred by participants. However, there is an important difference between their research and ours: *context of use* (aka *task*). Their user study employed a target-selection task. The user moved an object (using device tilt) from a home position to a destination target and selected the target by positioning the object inside the target. The goal was to select the target as quickly as possible. Our user study (described in the next section) uses a path-following task. The user manoeuvres a ball through a path that requires several direction changes. The goal is to navigate through the path without touching the path walls. This task is similar to a traditional marble-maze game, where the movement of the ball is determined by force of gravity with the goal of reaching a destination without incident. The ball velocity and acceleration are determined by the angle of the maze surface. It remains to be seen whether the superiority of position-control reported by Teather and MacKenzie [17] for target-selection will extend to path-following. If so, this would be an interesting finding, since marble-maze games, as implemented on mobile devices, consistently use velocity-control (in an attempt to mimic real-world physics).

It is worth mentioning as well that while most research using device tilt as an input control uses data from a built-in sensor – typically an accelerometer – some research on tilt input uses the device's camera combined with image processing to compute the device's position and movement [e.g., 3, 7, 12, 14, 18]. The main disadvantage in using camera data is the substantial processing required in converting camera images to device movement. The result is system lag. For games, this is a deal-breaker since game-inputs must be highly responsive. For this, the high data rate and low processing requirements of an accelerometer are essential.

II. Method

A user study was conducted to determine which order of control is easier to use for tilt-based game-play. The study also sought to determine which order of control is preferred by users.

A. Participants

Twelve participants were selected using convenience sampling [9, p. 177] from a pool of staff and students at the local university campus. There were 6 males, 6 females. The average age was 23.3 years (min 19, max 40). Participants were paid a nominal compensation fee for their time (approximately 1 hour). It was noted that all participants used a smart device, with 83.3% indicating they play games on their device. Of those that play games, 70% stated they had little experience with tilt-based games.

B. Apparatus (Hardware and Software)

The experiment used a Samsung *Galaxy Tab 10.1* running Google's *Android 3.1* (Honeycomb) operating system. See Fig. 2. The device has a 10.1 inch display with resolution of 1280 × 800 pixels and a density of 149 pixels/inch.

Fig. 2 Samsung *Galaxy Tab 10.1*.

The software was developed in Java using the Android SDK, with special focus on the `SensorManager` class. Tilt control was implemented using the device's built-in accelerometer, manufactured by InvenSense (http://invensense.com). The sensor was configured to operate at the "game" sampling rate. Sensor events occurred approximately every 20 ms, providing tablet *pitch* and *roll* data to the experiment software. See Fig. 3.

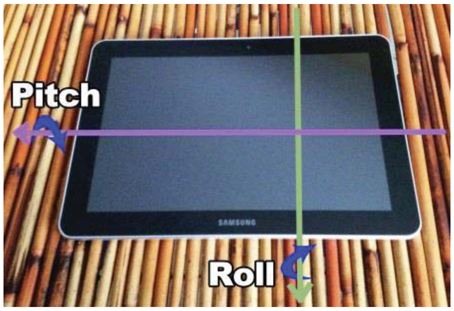

Fig. 3 Samsung *Galaxy Tab 10.1* (pitch & roll axis indicated).

The raw pitch and roll data were converted to aggregate measures of *tilt magnitude* and *tilt angle*, where tilt magnitude is the amount of tilt and tilt angle is the direction of tilt relative to an arbitrary direction (e.g., north). These were used to control the position or velocity of the ball, as now described.

Velocity-control mapped the tilt magnitude of the device to the velocity of the ball. The mapping was 100 pixels per second per degree of tilt, relative to flat. Obviously, the direction of ball movement was congruent with the tilt angle.

Position-control, on the other hand, directly mapped the device's tilt magnitude and tilt angle to ball position. In this case, the mapping was 20 pixels per degree of tilt magnitude in the direction of tilt angle. Setting the device flat on a table positioned the ball in the center of the display.

The mappings just described are equivalent to the "nominal" settings used by MacKenzie and Teather [10].

The experiment application was *TiltMaze*, a marble-maze-like game developed specifically for this research. *TiltMaze* uses device tilt in either velocity-control or position-control mode to manoeuvre an object – a virtual ball – on the display surface. The application was organized in three parts: setup, testing, and results. A setup screen was used to set parameters of interaction, such as a participant code and order of control. A results screen appeared after a block of testing to allow participants to view their performance. The testing section was a simple game for participants to play while their performance was measured. The objective of the game was to direct a virtual ball – the object under tilt control – from a green start circle (top left) to an opposing circle (bottom right) while manoeuvring the ball within and through a path. As seen Fig. 4a, the path included several changes in direction on the way to the opposing circle.

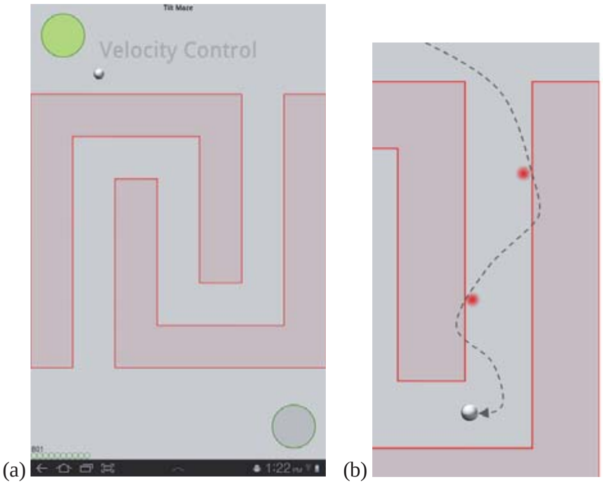

Fig. 4 *TiltMaze* (a) Screenshot of the game display (b) Markers showing wall hits (Note: Dashed line added to illustrate ball path).

If the ball veered off the path and hit a wall, this was indicated in three ways: (i) a visual marker appeared, (ii) a brief audio click was heard, and (iii) a 10-ms vibrotactile pulse was emitted. The visual marker is shown in Fig. 4b (dashed line added).

Timing for a trial began when the ball entered the start circle and ended when the ball entered the opposing circle. The data logged included the time for the trial, the number of times the ball hit a wall, the pixel distance over which the ball moved, and the amount of off-path time and off-path distance.

At the end of each trial, the participant continued immediately with the next trial which involved moving the ball back to the start circle (now the opposing circle). Ten trials were considered a block. At the end of a block, a popup results screen appeared showing the time, the number of wall hits, and the movement efficiency. Movement efficiency was calculated as the "nominal distance" divided by the "actual distance", with nominal distance defined as the distance from the one circle to the other moving through the center of the path.

C. *Procedure*

Participants were first informed about the nature of the study after which they were asked to sign a consent form. Next they were given instructions on the operation of *TiltMaze* and the task and goal. One block of 10 trials was demonstrated by the experimenter, after which the participant did a practice block with the first order of control. Participants then completed 10 blocks of testing during which their performance was measured with data logged in the device's internal memory. The data were later transferred to a host computer for analysis. Fig. 5 shows a participant performing the experiment task.

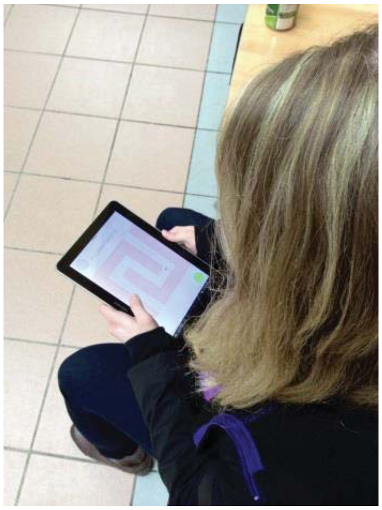

Fig. 5 Participant during testing.

For each trial, participants were asked to control the ball by tilting the tablet. They were informed that the objective was to remain within the path, avoid hitting the walls, and to proceed as quickly as possible. The same procedure was repeated for the second order of control. After testing, each participant completed a questionnaire to gauge their preference between the two orders of control as well as ease of use.

D. *Design*

The study was a 2 × 10 within-subjects design with the following independent variables and levels:

Order of control:	position-control, velocity-control
Block:	1, 2, 3 … 10

Each participant completed 10 blocks of 10 trials each for each order of control. Thus, the total number of trials for the experiment was 12 participants × 2 orders of control × 10 blocks × 10 trials/block = 2,400.

To offset learning effects, participants were divided into two groups to counterbalance the order of testing. One group used position-control first, then velocity-control. The order was reversed for the other group.

The dependent variables were time per trial, number of error-free trials, wall hits per trial, and movement efficiency. While these variables are quantitative, a questionnaire was also used to gather qualitative measures on user preference and ease of use.

III. Results and Discussion

Post-study analyses resulted in interesting findings across the two orders of control on the time per trial, the number of wall hits per trial, and the movement efficiency. The qualitative analysis also had interesting, yet contradictory, findings in terms of user preference and ease of use.

A. *Time Per Trial*

The time per trial was the main dependent variable. Over all 2,400 trials, the mean time per trial was 10.7 seconds. By order of control, the means were 9.93 seconds for position-control and 11.54 seconds for velocity-control. Thus, trials took on average 16.2% longer when participants were moving the ball using velocity-control. See Fig. 6. The difference was statistically significant ($F_{1,10} = 5.46$, $p < .05$).

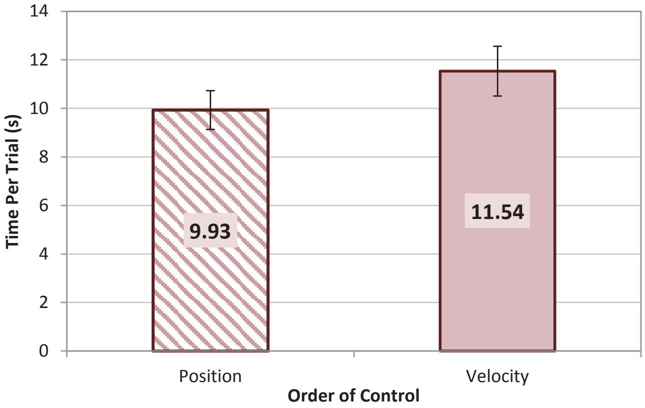

Fig. 6 Time per trial (s) by order of control. Error bars show ±1 *SE*.

This is an important finding since it provides the first empirical evidence for the viability of position-control over velocity-control for tilt-based input in path-following tasks. This result corroborates Teather and MacKenzie's similar finding for target-selection tasks [17]. The result is particularly interesting here since velocity-control tilt-input is the most common choice for path-following tasks in mobile games.

Over the course of the 10 blocks of testing, skill acquisition was evident as participants' time per trial decreased. This is shown in Fig. 7. Not surprisingly, the effect of block on time per trial was statistically significant ($F_{9,90} = 12.2$, $p < .0001$).

Models for the power law of learning are also shown in Fig. 7, accompanied with an extrapolation to the 20th block. Squared correlations are above .8 for both models, indicating the models are a good fit with the observed data. As blocks progress, there is a slight widening of the gap between position-control and velocity-control. This suggests that the speed benefit for position-control would continue as expertise develops further.

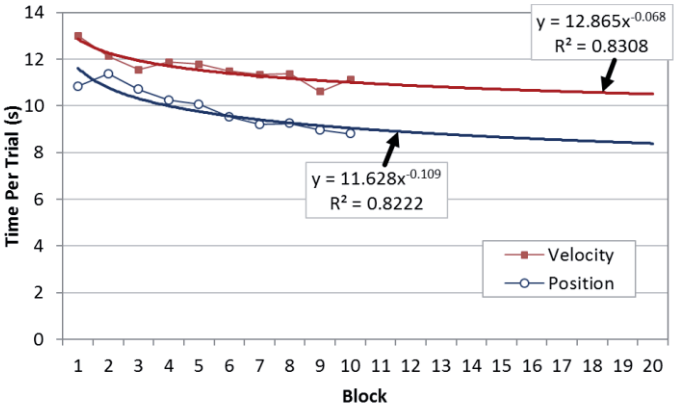

Fig. 7 Time per trial (s) by block with models for the power law of learning extrapolated to the 20th block.

B. *Error-free Trials and Wall Hits*

Over the entire user study, there were 1,200 trials each for position-control and velocity-control. The number of times users completed trials without the ball hitting a wall as 421 (35.1%) for position-control and 375 (31.3%) for velocity-control. Thus, there was a slight performance benefit for position-control over velocity-control in terms of error-free trials. The difference was not statistically significant, however ($F_{1,10}$ = 0.37, ns).

From the numbers above, it is clear that for most trials (>60%), there was at least one wall hit. In terms of the mean number of wall hits per trial, the values were 1.77 for position-control and 1.37 for velocity-control. See Fig. 8. It is interesting that even though there were more error-free trials with position-control, there were fewer wall hits per trial with velocity-control. However, this latter effect was also not statistically significant ($F_{1,10}$ = 0.31, ns), so caution is warranted on interpreting the results for error-free trials and wall hits per trial.

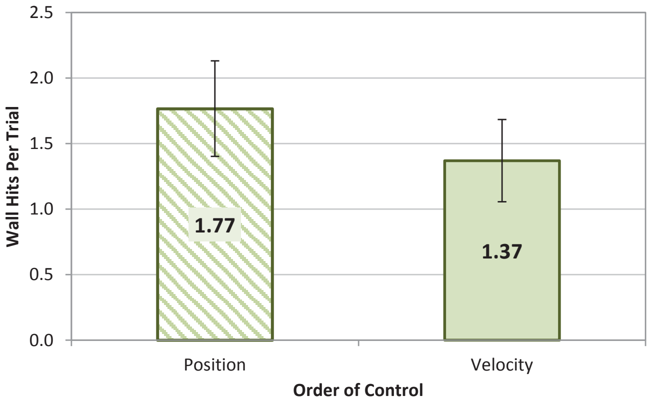

Fig. 8 Wall hits per trial by order of control. Error bars show ±1 *SE*.

The non-significant ANOVA for wall hits per trial is not surprising in view of the large error bars in Fig. 8, which suggest somewhat erratic behaviour among participants on their ability to control the path of the ball. Power laws of learning (not shown) were similarly erratic.

C. *Movement Efficiency Per Trial*

Movement efficiency reflects the tendency of participants to manoeuvre the ball along a direct path through the center of the available space. One reason movement efficiency data were gathered was to ensure participants did not cheat. If they moved the ball *through* the path walls directly from the start circle to the opposing circle (as permitted in the software), movement efficiency would be well above 100%. An inspection of the data indicated that this did not occur. Participants were clearly conscientious in attempting to perform well at the specified task.

The mean movement efficiency over all trials was 92.2%. By order of control, the means were 92.4% for position-control and 91.9% for velocity-control. See Fig. 9. Although this suggests a slight performance benefit of position-control over velocity-control, the difference was again not statistically significant ($F_{1,10}$ = 0.06, ns).

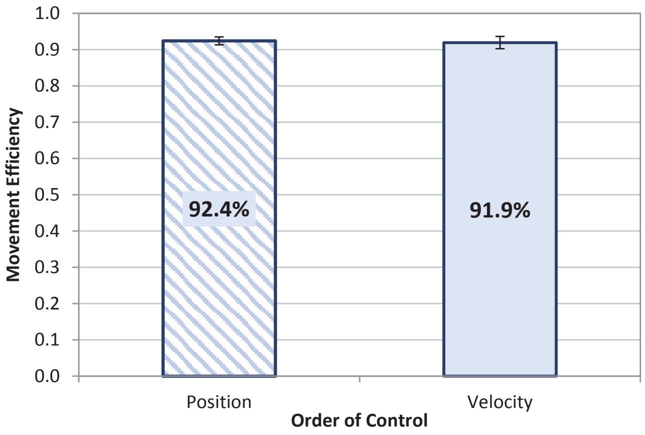

Fig. 9 Movement efficiency (%) per trial by order of control. Error bars show ±1 *SE.*

The learning pattern for movement efficiency was very different between position-control and velocity-control. This is shown in Fig. 10. Here, we see a reasonably smooth improvement in movement efficiency with position-control over the 10 blocks of testing. The extrapolation of the learning model to the 20th block shows that movement efficiency would likely continue to improve with position-control (R^2 = .6192). Bear in mind that with position-control, there is a specific progression of tilt angles required to manoeuvre the ball from the start circle to the opposing circle. Thus, there is the potential for participants to develop *motor memory* in the pattern of tilting. It is even conceivable that, with sufficient expertise, the task could be performed eyes free!

The same is not true for velocity-control. Adjustments in tilt alter the ball's velocity, rather than the ball's position. Thus, movements are, by necessity, feedback-guided: Tilt aggressively and the ball overshoots; tilt modestly and the ball plods along slowly. Participant transitions between aggressive and modest tilting are likely the cause of the erratic pattern of learning in Fig. 10 for velocity-control. This was also evident in the power law of learning for velocity-control (not shown) which had a low squared correlation (R^2 = .3341).

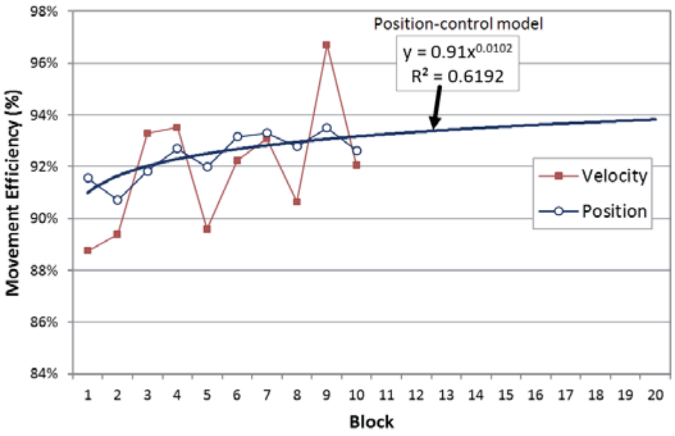

Fig. 10. Movement efficiency (%) by block. See text for discussion.

D. *Participant Preferences*

At the end of testing, participants were asked to respond to three questionnaire items on the orders of control used with *TiltMaze*. The first two items were

Rate how easy or difficult it was to use "Position Control"

Rate how easy or difficult it was to use "Velocity Control"

The responses were on a 5-point Likert scale ranging from Very Difficult (1) to Very Easy (5). The means of the responses were 2.67 for position-control and 3.08 for velocity-control. See Fig. 11. Although the ease-of-use rating was higher for velocity-control, the means for both orders of control are close to the neutral value of 3. Thus, neither order of control was rated as particularly easy. Furthermore, the difference in the responses was deemed not statistically significant using a Wilcoxon Signed-rank non-parametric test (z = -0.844, p > .05).

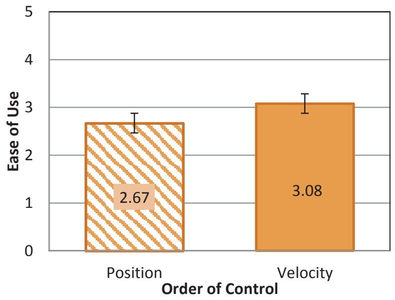

Fig. 11 Ease of use responses by order of control. Error bars show ±1 *SE*.

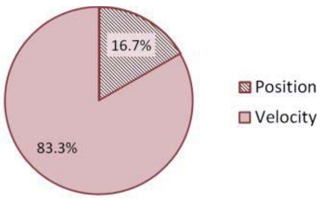

Fig. 12 User preference for order of control. Eight of 12 participants indicated a preference for velocity-control.

The third questionnaire item was a forced-response question asking participants to choose which order of control they preferred. Ten of 12 participants (83.3%) chose velocity-control. See Fig. 12. This result is in direct opposition to the fact that position-control took 16.2% less time per trial than velocity-control (and also had a slightly higher movement efficiency).

The questionnaire also included a box soliciting open-ended comments. Some participants provided explanations on their ratings and performance. It was noted by one that switching between the two orders of control "was extremely disorienting and difficult" [P11] and this caused a sense that scores had dropped when using the second order of control. Others felt that "position control consisted of jerky movements" [P08] which caused more wall hits than with velocity-control. At the same time, some participants felt that an inaccurate accelerometer was to blame for their inaccuracy [P02, P04]. One participant even mentioned that they "never got much better throughout the testing" when using velocity-control, however they preferred it due to its ease [P06]. The final thought most participants had during testing was that velocity-control had a more natural motion to it. This comment is a possible explanation for the preference that velocity-control received.

IV. Conclusion

At the commencement of this study, we were unsure whether position-control or velocity-control would prevail. Position-control might prevail, as reported by Teather and MacKenzie [17]. However, their study used a target-selection task. Conversely, velocity-control might prevail, since path following tasks (e.g., in marble-maze games) consistently use velocity-control. Three empirical measures were used in the comparison: one for speed (time per trial) and two for accuracy (wall hits per trial and movement efficiency). In the end, velocity-control only excelled in one of these three criteria: It had slightly fewer wall hits (although the difference was not statistically significant). Position-control excelled in speed by a significant amount: It was 16.2% faster. Movement efficiency was approximately the same between the two orders of control, although the learning progression for movement efficiency was much smoother with position-control. We consider these results important, meriting a re-think for designers of games using a path-following task. Although such games typically use velocity-control, position-control is a viable and potentially preferable alternative. Providing velocity-control *and* position-control is certainly an option designers should consider.

On user preferences, 83.3% of participants chose velocity-control, even though they performed better with position-control. This is also interesting. Their preference was perhaps due to the natural physics-like mapping when using velocity-control. The contradictory result is especially surprising given Teather and MacKenzie's finding that position-control was both more preferred by participants and afforded a better performance [17]. However, a few participants verbally stated during testing that the type and difficulty of the game also has an impact on which order of control they would prefer. Supplementary investigation with varying game types is required to validate and extend these participant claims.

Another verbal user comment was that the amount of tilt required for position-control was perhaps too much. Visual signals of frustration during testing confirmed this. A

"Medium" setting was used for gain when mapping tilt. If a more sensitive setting was used (higher gain), not as much tilt would be required to move the ball to the extremes of the display. This would speed-up the motions of the ball. Of course, the converse is that small amounts of tilt would move the ball farther than before, so using a higher gain setting for tilt could increase wall hits. Using different settings of gain is a potential independent variable for follow-up research.

Though surprising at first, the results make sense when considering that tablet movements have the possibility to use motor memory with position-control since the same movement always brings the ball to the same position. On the other hand, velocity-control does not afford as great a possibility for motor memory since the ball will keep rolling if one does not adjust quickly to prepare for the next direction the ball must go. Further study is required to determine the extent to which motor memory can be used with each order of control.

REFERENCES

[1] J. F. Bartlett, "Rock 'n scroll is here to stay," *IEEE Computer Graphics and Applications,* vol. 20, pp. 40-45, 2000 May/June.

[2] S. Boring, M. Jurmu, and A. Butz, "Scroll, tilt or move it: Using mobile phones to continuously control pointers on large public displays," in *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group - OZCHI '09*, New York:ACM, 2009. pp. 161-168.

[3] S. Bucolo, M. Billinghurst, and D. Sickinger, "User experience with mobile phone camera game interfaces," in *Proceedings of the 4th International Conference on Mobile and Ubiquitous Multimedia - MUM 2005*, New York: IEEE, 2005. pp. 87-94.

[4] D. Fitton, I. S. MacKenzie, J. C. Read, and M. Horton, "Exploring tilt-based text input for mobile devices with teenagers," in *Proceedings of the 27th International British Computer Society Human-Computer Interaction Conference – HCI 2013*, London: British Computer Society, 2013, pp. 1-6.

[5] P. Gilbertson, P. Coulton, F. Chehimi, and T. Vajk, "Using "tilt" as an interface to control "no-button" 3-D mobile games," *ACM Computers in Entertainment,* vol. 6, pp. 38:1-38:13, 2008.

[6] B. Harrison, K. P. Fishkin, A. Gujar, C. Mochon, and R. Want, "Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces," in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems - CHI '98*, New York: ACM, 1998, pp. 17-24.

[7] A. Henrysson, J. Marshall, and M. Billinghurst, "Experiments in 3D interaction for mobile phone AR," in *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia - GRAPHITE '07*, New York: ACM, 2007, pp. 187-194.

[8] K. Hinckley, J. Pierce, M. Sinclair, and E. Horvitz, "Sensing techniques for mobile interaction," in *Proceedings of the ACM Symposium on User Interface Software and Technology - UIST 2000*, New York: ACM, 2000, pp. 91-100.

[9] I. S. MacKenzie, *Human-computer interaction: An empirical research perspective*: Waltham, MA: Morgan Kaufmann, 2013.

[10] I. S. MacKenzie and R. J. Teather, "FittsTilt: The application of Fitts' law to tilt-based interaction," in *Proceedings of the 7th Nordic Conference on Human-Computer Interaction - NordiCHI 2012*, New York: ACM, 2012. pp. 568-577.

[11] S. Medryk and I. S. MacKenzie, "A comparison of accelerometer and touch-based input for mobile gaming," in *Proceedings of the International Conference on Multimedia and Human-Computer Interaction - MHCI 2013*, Ottawa, Ontario: ASET, Inc, 2013, pp. 117.1-117.8.

[12] J. Mooser, S. You, and U. Neumann, "Large documnet, small screen: A camera driven scroll and zoom control for mobile devices," in *Proceedings of the Symposium on Interactive 3D Graphics and Games - I3D '08*, New York: ACM, 2008. pp. 27-34.

[13] I. Oakley and S. O'Modhrain, "Tilt to scroll: Evaluating a motion based vibrotactile mobile interface," in *Proceedings of the Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems -- Eurohaptics 2005*, New York: IEEE, 2005. pp. 40-49.

[14] A. Ramsay, M. McGee-Lennon, G. A. Wilson, S. J. Gray, P. Gray, and F. De Turenne, "Tilt and go: Exploring multimodal mobile maps in the field," *Journal of Multimodal User Interfaces*, vol. 3, pp. 167-177, 2010.

[15] J. Rekimoto, "Tilting operations for small screen interfaces," in *Proceedings of the ACM Symposium on User Interface Software and Technology - UIST '96*, New York: ACM, 1996. pp. 167-168.

[16] D. Small and H. Ishii, "Design of spatially aware graspable displays," in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems - CHI '97*, New York: ACM, 1997, pp. 367-368.

[17] R. J. Teather and I. S. MacKenzie, "Position vs. velocity control for tilt-based interaction," in *Proceedings of Graphics Interface 2014 - GI 2014*, Toronto: Canadian Information Processing Society, 2014. pp. 51-58.

[18] J. Wang, S. Zhai, and J. Canny, "Camera phone based motion sensing: interaction techniques, applications and performance study," in *Proceedings of the ACM Symposium on User Interface Software and Technology - UIST '06*, New York: ACM, 2006, pp. 101-110.