



Tilt-Touch synergy: Input control for “dual-analog” style mobile games



Robert J. Teather^{a,*}, Andrew Roth^b, I. Scott MacKenzie^c

^a Carleton University, School of Information Technology, 1125 Colonel By Dr, Ottawa, ON K1S 5B6, Canada

^b Brock University, Centre for Digital Humanities, 1812 Sir Isaac Brock Way, St. Catharines, ON L2S 3A1, Canada

^c York University, Department of Electrical Engineering and Computer Science, 4700 Keele Street, Toronto, ON M3J 1P3, Canada

ARTICLE INFO

Article history:

Received 28 June 2016

Revised 11 April 2017

Accepted 22 April 2017

Available online 29 April 2017

Keywords:

Tilt control

Touch control

Dual-analog games

Shooter games

ABSTRACT

We propose the use of device tilt in conjunction with touch control in mobile “dual-analog” games – games using two virtual analog sticks to independently control player movement and aiming. We present an experiment investigating four control modes based on this strategy. These include a standard dual analog control scheme, two options using tilt control in lieu of touch control for either movement or aiming, and a tilt-only control scheme. Results indicate that while touch-based controls offered the best performance, tilt-based movement control was comparable. In contrast, tilt-based orientation control significantly altered, and in certain cases impaired, participant navigation. Overall, participants preferred touch-based control, but tilt-based movement with touch-based aiming was a close second in subjective preference.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Modern game controller design has largely coalesced around a configuration including two small analog joysticks, numerous face buttons, and two to four shoulder buttons. This style of joystick (on gamepads) first gained popularity in the late 90s as 3D platform games emerged [9]. Fig. 1 depicts standard controllers for the three largest game console hardware manufacturers. Note the high degree of consistency among these.

These complex controllers have yielded correspondingly complex game control schemes. Modern games often use both joysticks in tandem. We refer to these as *dual-analog* games. There are several examples, including first-person shooter (FPS) and top-down shooter games. Both genres typically use the left joystick to control player movement and the right joystick to control viewpoint or orientation. We focus on the latter category (top-down shooters).

Although there is some variation between controllers, typically the joysticks are positioned in areas accessible by the thumbs. The face buttons are generally positioned near the right joystick. Normally, it is difficult or impossible to simultaneously use the face buttons and the joysticks (with the same thumb), which has led to the proliferation of shoulder buttons (as many as four on modern controllers) accessible while using both joysticks.

Certain types of mobile games – notably ports and re-makes of console games for mobile devices – simulate the controllers described above via software. Such simulated controls are referred to as soft or virtual controls. Example virtual controls are depicted in Fig. 2.

Virtual controls have three notable deficiencies compared to their physical counterparts:

1. The absence of tactile feedback yields a noticeable performance cost [3,27,30], although it is argued that unpredictable touch sensor events may also matter [17]. Fingers can easily miss buttons and slide off virtual joysticks at crucial moments.
2. Like physical controllers, virtual face buttons are inaccessible while using the virtual joysticks. Unlike physical controllers, mobiles do not include shoulder buttons to circumvent this issue.
3. Virtual controls occlude portions of the screen and, hence, sometimes distract from gameplay.

Dual-analog games that use two virtual analog sticks are the most susceptible to the above problems.

There are hardware solutions to these problems. For example, connecting a Bluetooth controller to a mobile device offers a similar experience to console controllers. However, this necessitates extra hardware and is awkward in truly mobile scenarios. For example, consider the difficulty in holding both a tablet and a game controller while standing on a moving bus. Hardware developers have also designed their mobile game platforms

* Corresponding author at: 230G Azrieli Pavilion, School of Information Technology, Carleton University, 1125 Colonel By Drive, Ottawa, ON K1S 5B6, Canada.

E-mail address: Rob.Teather@carleton.ca (R.J. Teather).



Fig. 1. Modern video game controllers from major hardware manufacturers. (a) Sony PlayStation 4 controller, which includes a touch-sensitive trackpad; (b) Microsoft Xbox One controller which features asymmetric analog sticks, consistent with earlier designs; (c) Nintendo's Wii-U Pro GamePad features a resistive touchscreen and (d) Nintendo's Switch Joycon controllers, detachable controllers held in the Joycon grip.



Fig. 2. Rockstar's *Grand Theft Auto 3* mobile port includes virtual controls. The virtual analog joystick (left thumb) and virtual buttons (right thumb) are circled.

(e.g., Playstation Vita, Nintendo 3DS) to include shoulder buttons, but this requires purchasing a separate mobile device for games.

We instead argue for solutions that engage sensors that are ubiquitous in modern mobile devices, including accelerometers, gyros, magnetometers, and cameras. The principle advantage is that no additional hardware is required. Game/UI designers can instead focus on developing effective software mappings utilizing the sensor input. Tilt control, supported by accelerometers and gyros, is particularly promising. This is likely due to the naturalness of interaction offered by tilt and similar 3D gestures. Gamers are already familiar with tilt control, as it is commonly employed for viewpoint control in commercial console games (e.g., for aiming in first-person shooting scenarios, such as the bow used in *The Legend of Zelda: Breath of the Wild*). Mapping 3DOF sensor data (pitch, roll, and yaw) to 3D camera orientation is also common practice in AR/VR games available on recent platforms such as the Oculus Rift and HTC Vive. Moreover, previous work in this area [22,27] did not reveal statistically significant differences in performance between tilt and touch control. There is thus merit in further study of tilt as a game control option, as the question of performance relative to touch control is still open to debate.

Careful design of tilt-based control techniques could potentially offer performance comparable to touch control. Tilt control also avoids two of the problems with virtual touch controls noted above. Specifically, tilt does not occlude the view, and can be used in tandem with virtual face buttons. And although tilt, like touch control, does not offer tactile feedback, it may instead leverage proprioception [23], which can help compensate for the absence of tactile feedback that touch control is noted for [30]. It has also been shown to offer superior user performance to camera-based control [8].

Our work is motivated by potential performance and user experience differences in novel mobile game control schemes. Our goal is to determine which touch controls can be replaced by tilt input without adversely affecting user performance or experience. Previous work investigated tilt as a replacement for touch input in mobile games [5,7,11,27]; however, the games studied tended to be simplistic, and not representative of “real” games. For example, the games studied in previous work used only a single analog input control – e.g., tilt replacing a single virtual joystick, or controlling a single moving object (like a rolling ball). This is insufficient to control dual-analog games, which necessitate two simultaneous analog inputs. In these cases, a single tilt sensor could replace one virtual joystick or the other. However, it is unclear which option – movement or orientation – makes more sense to replace with tilt. There is little previous work exploring the idea of using both tilt and touch input in tandem, and existing work uses tilt primarily as an adjustment to or scale factor for touch-based control, rather than a control mode in its own right [1]. We thus explore the synergy between tilt and touch control, investigating the design space of input options that use both touch and tilt in tandem to control dual-analog games.

We present a study comparing performance of touch and tilt control using a custom-developed dual-analog game, a top-down shooter. This class of game is popular on mobile platforms, yet is relatively understudied. Like FPS games, top-down shooters use one joystick to move the player, and the other to simultaneously aim and shoot in the specified direction. Unlike FPS games, the viewpoint is centered above the player. We investigate game control by replacing one or both virtual joysticks with device tilt. This

represents, to our knowledge, the first systematic exploration of combining tilt and touch control (which are commonly provided or studied independently) in games.

2. Related work

2.1. Virtual controls vs. physical controls

Virtual controls are software widgets simulating the joysticks and buttons on physical gamepads. Examples are seen in Fig. 2. The virtual joystick on the left controls the player's movement through the environment, while the virtual buttons on the right perform actions. Although similar to physical controls, virtual controls perform worse in practice, likely due to the absence of tactile feedback. This deficiency has been studied in both game [30,31] and non-game [4,14,18] contexts. For example, previous work using a touchscreen capable of providing haptic feedback [4,14] revealed that physical keyboards *and* the tactile touchscreen offered significantly faster text entry, fewer errors, and improved error correction over soft keyboards without tactile feedback. Participants also strongly preferred the tactile touchscreen. The authors suggest that since tactile touchscreens are not yet widely available, vibrotactile feedback (device vibration) might be used instead. Work by Lee and Zhai [18] further suggests that other feedback mechanisms (e.g., auditory) can compensate for the absence of tactile feedback.

Mobile gaming research indicates that the absence of tactile feedback strongly impacts player performance [30,31]. Zaman et al. [31] found that players were significantly slower and died more frequently using virtual controls on an *iPhone* than when using the physical controls of a *Nintendo DS*. Their study used a commercial game available on both platforms. However, there are many hardware differences between the devices, and potential differences between the versions of the game used. Hence these differences likely introduced confounding variables. Follow-up work [30] addressed some of these limitations by sticking a small physical joystick and foam buttons on the screen surface, co-located with virtual controls. The physical controls improved performance significantly, but still did not perform as well as a physical game controller.

Several other researchers investigated the problem of tactile feedback. Chu and Wong [6] report that players almost unanimously preferred physical controls over virtual controls. Oshita and Ishikawa [24] report that clever use of touchscreens can actually offer *better* performance than physical gamepads. They developed a soft button-based control scheme for a fighting game. Rather than emulating a physical gamepad, numerous virtual buttons were used where each button issued sequences of gamepad button presses (effectively macros for in-game controls). They found that, while gamepad entry speed was faster, the UI was less error-prone. However, the input method used numerous virtual buttons, and thus occluded a large portion of the screen. Tilt input does not present this problem.

Research on touchscreen-based soft controls is fairly clear that these offer inferior performance relative to physical controls. We instead look to tilt control as a possible alternative or adjunct to touch control.

2.2. Comparing touch and tilt input

Tilt control has been long recognized in the HCI community as a promising input modality. Numerous studies evaluated tilt for various tasks, including point selection [20,25,28], text entry [26,29], and 3D object manipulation [13]. There is also a considerable interest in employing tilt control in games [1,6,7,10,11,22,27].

Tilt input may leverage proprioception, the sense of the relative position of the limbs. Proprioception in lieu of tactile feedback was an early topic of interest in the virtual reality community [23]. Much like touchscreen-based virtual controls, VR systems usually do not provide tactile feedback when interacting with objects. Mine et al. [23] found that proprioception can instead create interaction mnemonics and improve gesture-based object manipulation in VR. We expect that tilt control will also leverage proprioception, and hence may work well in tandem with touch control. Similarly, other work comparing touch and physical controls indicates that extra feedback mechanisms (e.g., auditory feedback) can help improve performance in the absence of tactile controls [18].

Despite proprioception, tilt input is still limited since it mainly uses less dexterous joints such as the wrists [2,32]. In contrast, touch control employs the more dexterous finger/thumb muscles. Nevertheless, researchers report that tilt control offered faster text entry than multi-tap [29] and faster 3D rotations than touch control [13]. Unfortunately, there is relatively little research comparing touch and tilt input for games. Previous work is often contradictory and offers no strong take-away message. It is therefore possible that the opportunities for tilt may be highly task dependent. We thus isolate the use of tilt control both for player movement and player orientation/aiming, to assess which (if either) of these maps more naturally to tilt control.

There are a few examples of studies directly comparing touch and tilt input for games [3,5,15,22]. Browne and Anand [5] report participants preferred and performed better with tilt input than touch control, perhaps due to the inferior touch control implementation which did not support multi-touch. Medryk and MacKenzie [22] report that tilt offered worse performance than touch in a commercial game. However, using a commercial game in research limits one's ability to explore the design space of control options [16,21].

Teather and MacKenzie [27] compared order of control (position-control vs. velocity-control) across tilt and touch input in a Pong-like game. Interestingly, order of control had a greater impact than input method: Position-control offered significantly better performance than velocity-control for both tilt and touch input. There was no significant difference between tilt and touch. Later results [7] on tilt-control in a marble maze game confirm the better performance of position-control.

Of all studies on tilt and touch control for games, only Hynninen [15] and Alankuş and Eren [1] looked at complex dual-analog games, like our current work. Hynninen [15] reported that tilt input was inferior to virtual joysticks in a commercial game. However, the internal validity of this study is limited, since, like most commercial games, implementation details are unavailable. Alankuş and Eren [1] used tilt to supplement touch control, where device tilt controlled the rotation sensitivity of the second joystick. They reported that tilt-based aiming performed worse than the standard dual-analog joystick control and the tilt-augmented variant. Their tilt-augmented scheme performed worse than physical controls. While they focused exclusively on physical controllers, we instead look at touchscreen-based devices which may benefit more from the use of tilt control.

To address the limitations of previous work, we used a custom-developed game. This offers high internal validity, as it enables precise measurements. The game is also reasonably complex and potentially more representative of "real" games than the relatively simple games used in previous work [5,7,27]. Recreating the experience of a real game increases external validity by decreasing participant boredom or distraction during repeated testing [19]. Hence we expect that the external validity of our work is also high – the game is challenging, engaging, and should generalize to real games. Unlike most studies cited above, our game also employs dual-analog control. This offers a substantially more complex play

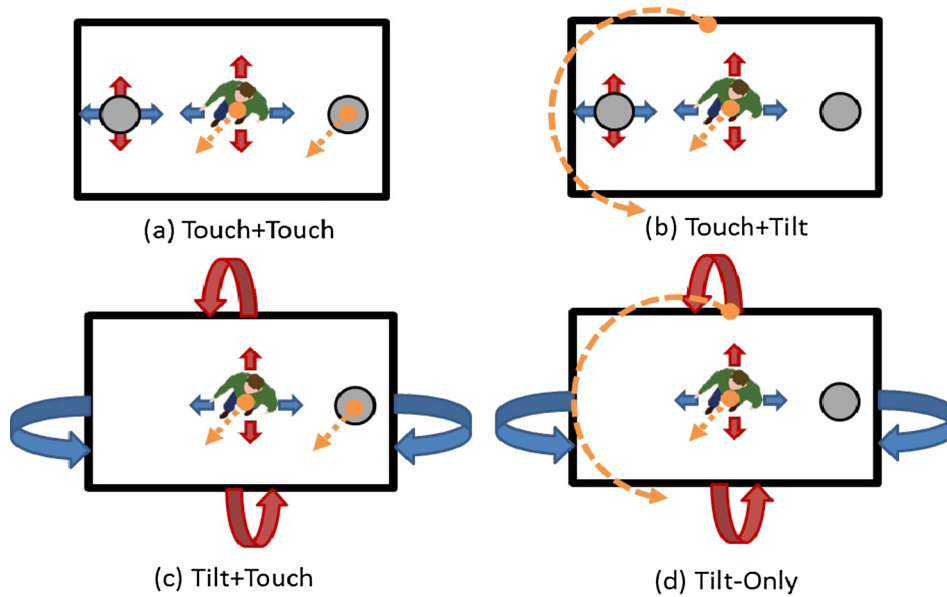


Fig. 3. Control modes studied – solid (red and blue) arrows indicate control of player movement, dashed (orange) arrows indicate control of player orientation. Curved arrows indicate device tilt. Gray circles indicate virtual analog sticks. See text for descriptions of each control mode. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Control mode descriptions.

Control mode	Player movement	Player orientation	Shooting
Touch + Touch	Left virtual joystick	Right virtual joystick	Coupled with orientation control
Touch + Tilt	Left virtual joystick	Device tilt (position-control)	Right joystick (as button)
Tilt + Touch	Device tilt (velocity control)	Right virtual joystick	Coupled with orientation control
Tilt-only	Device tilt (velocity control)	Device tilt (position-control)	Right joystick (as button)

Note: Movement and orientation coupled

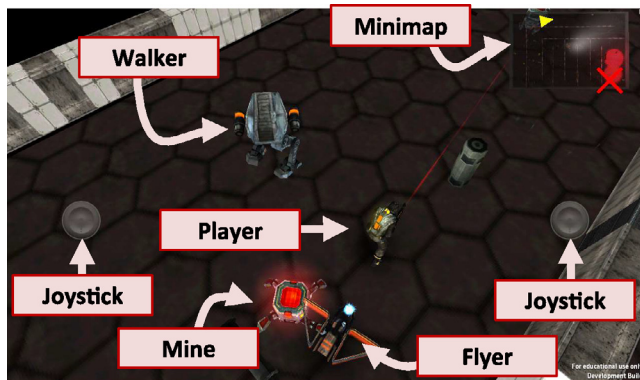


Fig. 4. Angry Bots Arena. The player stands near the centre of the screen. Enemy robots are near the player. A minimap was added at the top-right of the screen. Two virtual analog sticks are shown at either side of the screen.

experience than games used in most previous work. We chose a top-down shooter because the two rotational degrees of freedom provided by the gyroscope map well to the x and y axes of the controller while eliminating the confounding variable of the third axis.

3. Tilt-Touch synergy

Previous research has not reached consensus on the relative performance of tilt and touch control. While the baseline control

mode uses two virtual joysticks to independently control player movement and aiming direction, either or both joysticks could be replaced by tilt control.

This yields the four control modes explored in our study. See Fig. 3. The first three use a two-part naming scheme where the first term indicates how player movement is controlled and the second term indicates how player orientation is controlled. For example, Touch + Tilt indicates that player movement is controlled by a virtual joystick (touch) and player orientation is controlled by tilt. This naming scheme reflects the standard control mode, where the left virtual joystick controls movement, and the right virtual joystick controls orientation. Descriptions of the four control modes are summarized in Table 1.

The default, or baseline, control mode (Touch + Touch) used the two virtual joysticks seen in Fig. 4. The left virtual joystick controlled the player’s movement using a velocity control mapping that moves the player faster the farther from centre the virtual analog stick is pushed. Movement with all control modes (whether touch or tilt) always used a velocity-control mapping.

In contrast, orientation with all control modes used a 1:1 position-control mapping. Hence when using touch-based orientation control (e.g., Touch + Touch), touching the top of the virtual joystick aimed the player north; touching the right side of the virtual joystick aimed the player east. Similarly, tilting the device away from the player (rotating the tablet about the “right” or x-axis) in tilt-based orientation control aimed the player north. Tilting the device to the right (rotating the tablet about the “forward” or z-axis) in tilt-based orientation control aimed the

player east. Since the orientation mapping was 1:1 in both cases, all in-between angles (e.g., diagonal player orientations) were possible too.

With Touch + Touch and Tilt + Touch (i.e., touch-based orientation control), pressing the right virtual joystick also acted as a shoot button. The player would simultaneously shoot in the direction of orientation. In control modes that used tilt for orientation (i.e., Touch + Tilt, and Tilt-only), the right virtual joystick instead served as a shoot button (but did not affect player aim). This ensured that touch control was required for shooting with all control modes.

For tilt-based control modes, we adjusted the neutral or resting point of the tablet to 45° (in the *x*-axis), and 0° in the *z*- and *y*-axes. This corresponds to a similar viewing angle to that preferred by users with the standard Touch + Touch control mode. To avoid spurious activation of tilt control, we included a dead zone of 5° in any rotation axis away from the neutral position. Tilting the tablet beyond this range activated the tilt control for that axis of rotation, affecting player movement or orientation as described above. When tilt was used for movement, the maximum range of tilt (for full speed with the velocity-control mapping) required tilting the device by up to 30° in the desired movement direction. This value was determined through pilot testing to offer a comfortable means of controlling the tilt-based movement modes.

Interestingly, Touch + Touch (the standard control mode) conforms well to Guiard's model of bimanual control [12]. According to the model, bimanual tasks often use the non-dominant hand to coarsely set the frame of reference within which the dominant hand operates, performing fine control. With Touch + Touch, the left thumb (non-dominant hand for right-handed players) moves the character, and high precision is not generally required. The right thumb (dominant hand for right-handed gamers) requires greater precision and works within the framework established by moving the player. This may explain the standard control mode employed in dual-analog games, though the analogy breaks down for left-handed users. It is unclear if this analogy applies to tilt-based control modes.

An option we do *not* explore here is employing tilt to complement *two* touch controls. For example, one could use the Touch + Touch control mode but employ tilt as a tertiary control, such as increasing aiming sensitivity like Alankuş and Eren [1]. We may revisit this option in future work.

4. Methodology

We conducted a user study comparing the control modes in Fig. 3 (see also Table 1).

4.1. Participants

Sixteen paid participants (13 male) took part in the study. Ages ranged from 19 to 34 years (*mean* 22.6, *SD* 4.1). All were regular gamers and reported weekly use of games using either dual-analog physical game controllers or mobile virtual dual-analog controls. Only one participant reported they were left handed. Experienced participants were chosen to reduce result variability.

4.2. Apparatus (Hardware, Software, Gameplay)

The experiment used a Samsung *Galaxy Note 10.1* tablet with Google's Android 4.1.2 (Jelly Bean) OS. The display resolution was 1280 × 800 pixels and measured 260 mm (10.1") diagonally. Pixel density was 149 pixels/inch.

The game, *Angry Bots Arena*, is a heavily modified version of the tutorial included with Unity 4.5 (Fig. 4). The game uses a top-down

view and two virtual dual-analog joysticks that are functionally comparable to popular games. We positioned these virtual controls at the left and right midpoints of the screen to more closely align the tilt and touch conditions by balancing the device weight.

Angry Bots Arena involves controlling a player (shown near the centre of Fig. 4) to navigate a scene while avoiding or destroying enemy robots. The scene has a single, flat, and clearly delineated path to its terminus whereupon the player encounters a large boss robot. A game trial ends when the boss robot is destroyed. To quantify the frequency with which the player ran into walls (see environmental collisions in the Design section), the walls were broken into colliders about the same size as player.

Enemy robots came in three varieties, as shown in Fig. 4:

- Mines: Small robots resembling a trashcan that activate and chase the player, exploding on contact
- Flyers: Flying robots that approach the player and attack with a short range weapon
- Walkers: Tall robots that shoot small homing missiles.

All enemies except the boss were destroyed by two shots. This reduced the chances of an accidental kill at a distance, necessitating higher precision. The boss required more hits to defeat and more skill to avoid. Robot motion paths were constrained by bounding boxes to reduce randomness in each trial. The robots caused a range of damage to the player, from 5 points (for projectiles), up to 20 points (for Mines exploding at close proximity). The game was modified so the player had 20 health units, an infinite number of lives, and would immediately re-spawn at the same position upon dying. A brief white flash was used to indicate death rather than an animation that rewards or punishes dying. Player deaths therefore indicate performance independent of game time or player navigation skill.

Other software modifications were added to instrument the game for data collection (including six dependent variables; see below), wayfinding aids, and the four control modes. Two wayfinding aids were added to prevent participants from getting lost: a minimap (see Fig. 4, top right) and moving directional arrows that periodically appear on the floor. These were added to reduce navigation variability.

The four control modes described earlier (Fig. 3, Table 1) were compared, Touch + Touch, Touch + Tilt, Tilt + Touch, and Tilt-only.

4.3. Procedure

Upon arrival, participants were greeted by the experimenter, who explained the purpose of the study. They gave informed consent before starting. Participants were told to navigate the level quickly, shoot as many robots as possible along the way, and defeat the boss at the end. Participants then played through one practice trial using Touch + Touch, which was expected to be most familiar.

The experiment involved playing through the same map four times with each of the four control modes. The map is seen in Fig. 4 (top right) and in Fig. 7. The player started at the southwest corner of the map, and navigated through the scene to the end where the boss was located. Upon completing a trial, a short animation played giving players a short, consistent rest period, before the next trial began. It typically took a little over two minutes to complete a trial.

4.4. Design

The experiment employed a 4 × 4 within-subjects design. The independent variables and levels were as follows:

- Control mode (Touch + Touch, Touch + Tilt, Tilt + Touch, Tilt-only)
- Trial (1, 2, 3, 4).

Control mode ordering was counterbalanced with a balanced Latin square to offset learning effects. There were six dependent variables:

- Game time (seconds)
- Accuracy (percentage of shots that hit enemies)
- Enemies destroyed (count of enemies destroyed by the player)
- Player deaths (count of times the player was killed by enemies)
- Environment collisions (count of how often the player ran into walls or obstacles)
- Enemy collisions (count of how often the player bumped into enemies).

The software also recorded player position (x, y, z position), orientation (angle about the y -axis), device tilt (Euler angles about each of the three axes), and touch positions (both as offsets of the left/right joysticks, as well as x/y screen coordinates). Samples were collected at 20 ms intervals.

Subjective impressions were also gathered after testing via a six-item questionnaire with responses on a 5-point Likert scale.

5. Results and discussion

With two independent variables and six dependent variables, many analyses are possible. The six dependent variables were chosen to detect different performance characteristics of the control modes. Each of these represent a different aspect of the overall game play task. Specifically, game time was intended to assess if any control mode offered a speed advantage over the others, suggesting better usability. Accuracy was intended to detect differences in fine-grain targeting, while enemies destroyed would detect a cumulative difference in accuracy over a trial – both relating to the effectiveness mainly of the orientation control. The remaining three dependent variables – enemy collisions, player deaths, environment collisions – primarily provide insight into movement control. They indicate how readily the player could avoid getting hit by enemies, and dying because of it, while also smoothly navigating through the environment (i.e., not crashing into walls), respectively.

For the independent variable Trial, the most interesting result was for environment collisions (Fig. 5). No other dependent variable revealed significant effects for the Trial \times Control Mode interaction effect, suggesting fairly consistent improvement across all trials for each control mode. We thus only report the results for environment collisions. See Fig. 5. Note the relatively similar and flat responses (i.e., little to no improvement over trials) for all

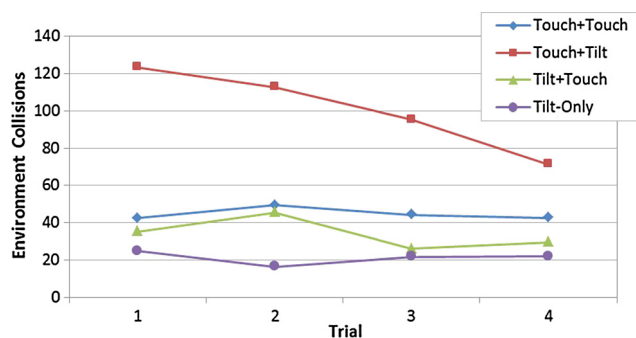


Fig. 5. Environment collisions by Control Mode and Trial.

control modes except Touch + Tilt, which fared much worse, with many more environment collisions. Yet, there was substantial improvement over the four trials. With continued practice, perhaps Touch + Tilt would demonstrate movement control comparable to other control modes. Not surprisingly, the effects were significant for Trial ($F_{3,45} = 6.72$, $p < 0.0001$), Control Mode ($F_{3,45} = 14.7$, $p < 0.001$), and Trial \times Control Mode interaction ($F_{9,135} = 3.51$, $p < 0.001$).

Since Trial had relatively little impact on the results, we average all four trials for subsequent analyses for each dependent variable. The results by Control Mode for the six dependent variables are gathered together in Fig. 6. The figure provides a bar chart and the ANOVA result for each dependent variable, averaged over all four trials. Where the ANOVA was significant, the control mode yielding the best performance is underlined in red. As well, horizontal bars (●—●) indicate significant pair-wise differences, as determined in a Bonferroni-Dunn post hoc comparisons test ($p < 0.05$).

The most interesting observation in Fig. 6 is that the best performing Control Modes are Touch + Touch and Tilt + Touch, followed by Tilt-only, each of which was best on at least one dependent variable. Touch + Tilt was consistently worst. Since each dependent variable exposes a different aspect of play (e.g., shooting enemies, avoiding enemies, not colliding with walls, etc.), it suggests that different styles of play are better supported by different control schemes. This is discussed further below. We now summarize the results for each dependent variable.

5.1. Game time and accuracy

Game time is mean duration of a trial in seconds. The Touch + Touch control mode was the most expeditious with participants taking on average 141 s per trial. The slowest method was Tilt-only at 164 s per trial.

Accuracy is the ratio of shots that hit the enemy to all shots. A statistically significant effect did not appear on accuracy. For all control modes the accuracy ratio was about 0.2, implying that about 1 in 5 shots hit the enemy.

Notably, Tilt + Touch was not significantly slower and had a comparable game time to the best performing control mode, Touch + Touch. In fact, game time scores for both Tilt + Touch and Touch + Touch were significantly better (lower) than the other two control modes. However, Tilt + Touch had the worst accuracy, although this was not significant. Given that the accuracy scores were all quite close, this suggests there is some promise for tilt-based movement. On average, Tilt + Touch was comparable to Touch + Touch in terms of game time and accuracy.

5.2. Enemy collisions and environment collisions

Enemy collisions is the number of times per trial the player bumped into an enemy. The goal, of course, is to shoot the enemy, not collide with the enemy. The Tilt-only control mode was significantly worse than the other control modes, registering, on average, 20.0 collisions per trial. With each trial taking about two minutes, this equates to about 1 collision every 6 s. The Tilt + Touch control mode fared best, averaging only 7.9 collisions per trial.

Environment collisions is the number of times per trial the player bumped into a wall or obstacle along the course. If a user employed a strategy of avoiding damage or obstacles by sliding along a wall, it is reasonable to assume their strategy would not differ between control schemes. A variance between control schemes implies that environmental collisions are a result of user error. This is validated by the motion path data (discussed below). The Touch + Tilt condition fared worst, averaging 100.8 collisions

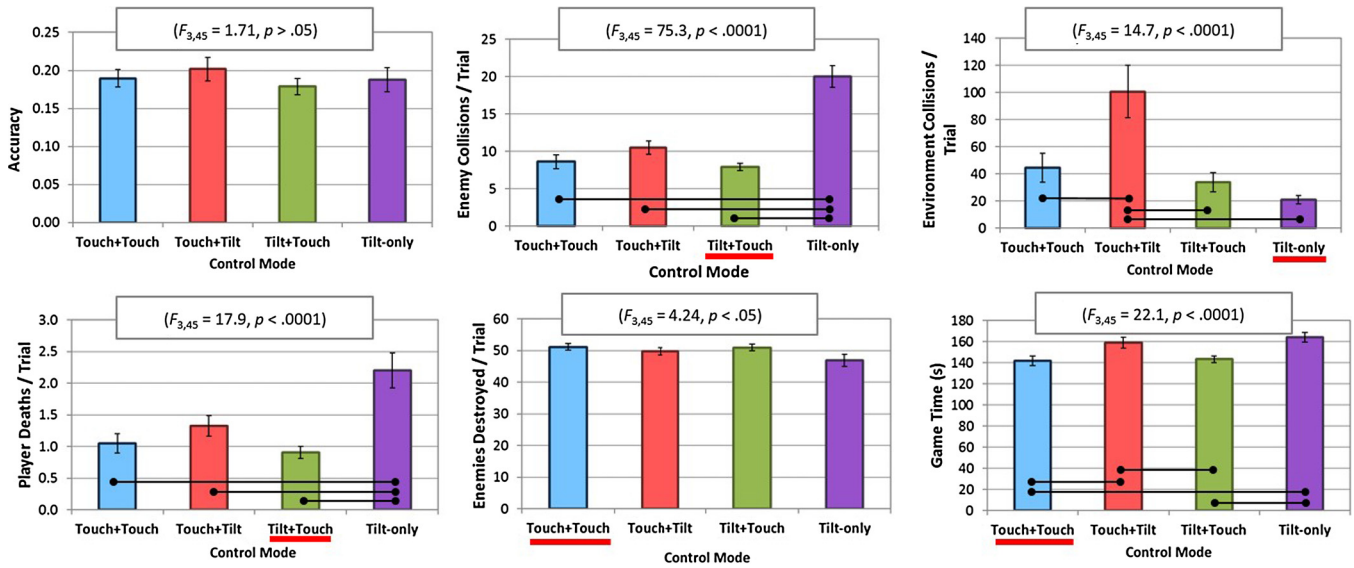


Fig. 6. Results for six dependent variables by Control Mode. See text for discussion. Error bars show ± 1 SE.

per trial. The best control mode was Tilt-only with an average of 21.2 collisions per trial.

Both enemy collisions and environment collisions indicate how easily participants could control their movement. Interestingly, the worst performer was different for these two metrics. For enemy collisions, Tilt-only was worst, while Touch + Tilt was worst for environment collisions – and yet Tilt-only offered the best performance here! In contrast, participants had considerable difficulty with Touch + Tilt, which had the highest environment collision score. See the Motion Paths section for further detail.

5.3. Player deaths and enemies destroyed

Player Deaths is the number of times per trial the player was killed by enemies (bearing in mind that the player re-spawns when killed). Obviously, lower scores are better. The Tilt + Touch control mode performed best, with an average of 1.1 deaths per trial. The worst condition was Tilt-only, with an average of 2.2 deaths per trial. This is likely related to the high number of enemy collisions with Tilt-only – colliding with enemies damaged the player, especially the Mine robots that exploded on contact.

Enemies Destroyed is the number of enemies killed per trial. Higher scores are better. Although the ANOVA revealed a statistically significant outcome ($p < 0.05$), no significant pairwise effects were found by the Bonferroni-Dunn test. The Touch + Touch condition had the most enemies destroyed, with an average of 51.1 per trial. However, Tilt + Touch virtually tied this score at an average of 50.9 enemies destroyed/trial. Based on the scores, and despite the absence of significant pairwise effects, it appears that Tilt-only was slightly worse than the rest. All other control modes had similar scores. It is interesting that Touch + Tilt, which offered significantly worse navigation control offered comparable performance in terms of enemies destroyed to the better performing conditions. This, coupled with its comparable (but not significantly better) accuracy suggests there is merit to our earlier suggestion that tilt-based control may effectively leverage proprioception. Due to the position-control mapping, participants tilted the tablet in an enemy’s direction to aim. This apparently offered a natural aiming scheming, despite the problems it presented to navigation, as reflected by environment collisions.

5.4. Motion path

Fig. 7 depicts player motion for typical trials with each control mode. These figures show how players progress through the level and indicate both player position and orientation using directional cone glyphs. The tip of the cone therefore aligns with the heading of the player.

The player’s orientation is also mapped to color values using the color wheel shown in the legend. The CMYK color wheel is useful for delineating between gradual and sharp changes in orientation throughout the trial. For example, it is evident from Fig. 7b (Touch + Tilt) that the participant oriented the character towards the upper-right direction almost throughout the entire trial (green, cyan). With this control mode, participants often ran sideways, employing tilt-based aiming minimally. Despite the virtual joystick for movement, they tended to slide along walls more often, yielding a significantly higher environment collision score (see also Fig. 6). This suggests tilt-based aiming adds cognitive load to participants’ movement task.

As discussed, this reveals that tilt-based orientation control had an unexpected side-effect on navigation. In contrast, Fig. 7a and c reveal that the participant periodically turns around to shoot at enemies (depicted as “spiky” parts along the otherwise fairly smooth path). The path is smoother in Fig. 7a than Fig. 7c, likely due to differences in touch and tilt-based movement control.

Fig. 7d depicts Tilt-only. Participants performed long smooth paths with minimal turning avoiding walls as much as possible, yielding the lowest environment collisions score (see Fig. 6). There are fewer attempts to spin around to shoot enemies (spiky parts). This is likely because of the inherent difficulty in maintaining quick navigation without independent aiming and movement directions available to other control modes. This is further supported by the lower number of enemies destroyed and higher game time (see Fig. 6). Tilt-only also had by far the most enemy collisions, but accuracy was not significantly worse than the other control modes. Based on the lower number of enemies destroyed, this likely indicates that participants simply did not even attempt to shoot as many enemies. Overall, these results suggest that coupling aiming and movement motivates the user to run through enemies while shooting, despite enemy collisions. These considerations will likely influence future studies, as discussed below in Section 6.1.

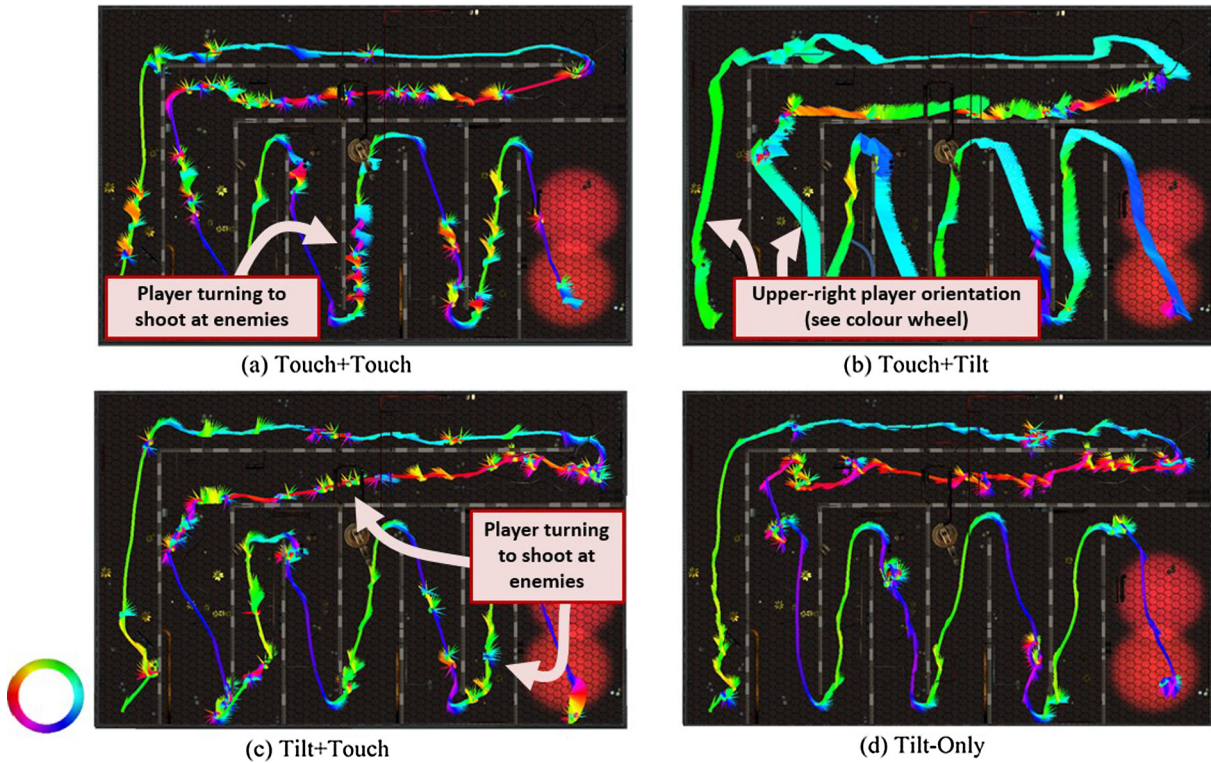


Fig. 7. Position and orientation of player in using cone glyphs in each of the four control modes (a) Touch + Touch. (b) Touch + Tilt. (c) Tilt + Touch. (d) Tilt-only. Orientation legend is a CMYK color wheel (bottom left), representing a 1:1 mapping of player orientation in the figure. For example, a bright green arrow arrow/line indicates the player was pointing “North”, while a dark blue arrow/line indicates the player was pointing South. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.5. Subjective results

After the experiment, participants were surveyed for their subjective impressions using items on a 5-point Likert scale. They were asked to rank how smooth the control was with each control

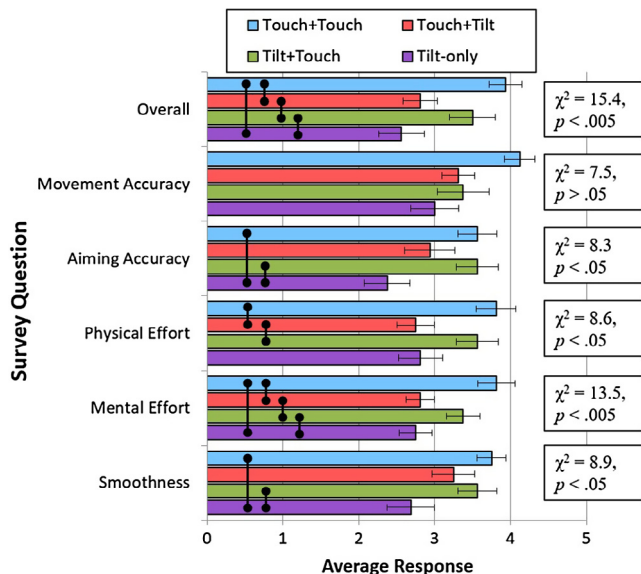


Fig. 8. Subjective results for each control mode by survey question. Higher scores are better. Error bars show ±1 SE. Statistical results via the Friedman test shown to the right. Vertical bars (●●) show pairwise significant differences per Conover's F test posthoc at the $p < 0.05$ level.

mode, the mental and physical effort required, how accurately they could aim and move, and their overall impression. Results are summarized in Fig. 8. Higher scores are better.

Participants tended to prefer Touch + Touch over the tilt-based control modes. However, Tilt + Touch also fared well and was consistently ranked second after Touch + Touch. This generally supports our data, which suggested that tilt-based movement with touch-based aiming tended to offer better performance than the inverse control mode (Touch + Tilt). Overall, participants significantly preferred both Touch + Touch and Tilt + Touch over Tilt-only and Touch + Tilt. In terms of aiming accuracy, they significantly preferred Touch + Touch and Tilt + Touch over Tilt-only (likely due to the movement/orientation coupling). Across most of the questions, Tilt + Touch fared substantially better – significantly so in many cases – than Touch + Tilt and Tilt-only. Interestingly, one participant stated that they found “Touch + Touch the easiest, but Tilt + Touch the most engaging and fun” control mode. Another stated they felt Touch + Tilt offered a good balance between the degree of tilting required and control. These results suggest that participants may be more accepting of tilt-based control modes, should they offer comparable performance levels. That said, one participant also noted higher degrees of arm fatigue when using the tilt-based control modes.

6. Discussion and design guidelines

While the tilt-based control modes did not outperform Touch + Touch, in some cases their mean scores were close, and our data failed to detect a significant difference between them. In the case of Tilt + Touch, this tentatively suggests that almost equivalent performance can be achieved while eliminating a second virtual joystick. Hence, we suggest further study of control modes merging

tilt and touch. We now summarize and discuss our most salient findings.

First, navigation was more difficult with touch control than with tilt control. This is reflected in the comparatively good performance of the Tilt + Touch in environment and enemy collisions, and Tilt-only in environment collisions. We argue that the comparatively worse performance of Tilt-only in terms of enemy collisions (and consequently, player deaths) is a side effect of the absence of independent aim/movement control with that control mode, rather than an inherent deficiency of tilt control itself. Note that the coupling of aiming and movement direction was unique to Tilt-only. The inability to aim independent of the movement direction made it difficult to precisely target enemies during movement, and thus, participants tended to run through the enemies rather than destroy them. Similarly, game time suffered as participants had to “loop” around enemies in order to destroy them. Given that both control modes employed tilt for movement control we argue that taken together, this supports our suggestion that tilt makes more sense for navigation than aiming.

In contrast, the one control mode which used tilt for aiming (Touch + Tilt) was generally the worst performing and least preferred condition, particularly if one excludes Tilt-only due to its absence of independent aim/movement control (which is a clear handicap for that condition, and effectively biases the results against it). This result was surprising: in developing the control modes, our personal expectation was that Touch + Tilt would be competitive with, and perhaps better than Tilt + Touch. We initially expected Touch + Tilt, a 1:1 mapping of tilt direction to player orientation, to serve as a natural and hence effective control mode. Since the player avatar was always in the screen centre, participants simply had to tilt the device in direction of an enemy relative to the centre of the tablet. One possible explanation for these results is that tilting the device far from the neutral orientation made it difficult to see where one was aiming (and navigating); however, this explanation is inconsistent with our other results. Touch + Tilt did not necessitate a great degree of tilting – the 1:1 mapping ensured that simply tilting the device out of the neutral “dead zone” at all would instantly orient the player to face the corresponding direction. In fact, the tilt-based movement control modes actually required a greater degree of tilting, since they included the ability to control the character speed by tilting slightly farther (like an analog joystick). Consequently, if extreme tilt angles were the culprit here, that should reflect more strongly in the tilt-based movement control modes rather than with tilt-based aiming. We would also expect to see more pronounced orientation corrections in the Touch + Tilt motion paths if the user were trying to centre the device in a way that affected performance. We thus argue that the worse performance of tilt-based aiming was *not* a result of extreme tilt angles in the Touch + Tilt condition. The motion paths do indicate that the character’s orientation and navigation were less congruent with Touch + Tilt than in the other conditions so it is likely more complex phenomena are at play.

Given the asymmetry in bimanual control noted by Guiard [12], it is possible that leveraging proprioception in the manner of Touch + Tilt yields an unnatural mapping. For example, Guiard’s model indicates that the non-dominant hand sets the spatial frame of reference and performs coarse-grain control (e.g., moving the player avatar), while the dominant hand performs fine precision tasks within that frame of reference (e.g., aiming at enemies). However, there are two layers to consider in our study – setting the spatial frame of reference in the game, versus that of the tablet itself. Given that almost all our participants were right handed, we expected that using the left virtual joystick to control player movement would offer a natural mapping. This clearly sets the in-game

spatial frame of reference per Guiard’s model, while aiming would constitute a fine precision task. But it is unclear if tilting the tablet itself (setting a different spatial frame of reference) is a dominant or non-dominant hand task – or neither. Hence it is possible that the non-dominant hand may simply have been overloaded, being simultaneously responsible for controlling two different spatial frames of references.

An alternative explanation is that despite the presence of a neutral dead zone (within which tilt was ignored), as noted in Fig. 7, participants tended to move the player avatar through the environment sideways, indicating that the device was at least slightly tilted out of the dead zone throughout the entire experiment. Hence to aim in that condition, they might have had to periodically tilt *through* the dead zone to aim at enemies on the opposite side of their unintended default orientation. In that case, tilting towards the intended aiming direction would require a larger motion than we otherwise expected (potentially greater than the 10° total range of the dead zone in either axis). In contrast, in the conditions using tilt for movement, any deviation from the dead zone resulted in undesired movement, which had an immediate, obvious, and continuous effect on the game – the player avatar would actually move in an undesired direction. In contrast, navigating through the maze sideways only mattered when trying to shoot enemies. Hence movement control required greater participant attention. Participants might have been less concerned about their default tilt direction when used with orientation (the cost of such an error was apparently low), which in turn may have a subtle influence on performance.

A final possible explanation for the superiority of tilt for movement over orientation is the naturalness of such a mapping. We had initially surmised that proprioception might play a role and tilting towards an enemy to aim at them is a natural gesture. However, one could argue that tilting to move an object along a surface is a similarly natural and potentially even more intuitive gesture. Of course, this is exactly the control style used in marble-maze games. It simulates what happens in the real world when tilting a physical object with another object resting on its surface; the resting object will slide or roll in the direction of the tilt. We have an intuitive understanding of such a mapping from early childhood, as we being to understand how physics operate in the world. This might explain the performance differences observed in our study.

With additional practice, some control modes might improve further, at least for some performance metrics (e.g., see Fig. 5). Notably, the only control mode that showed any signs of improvement over the course of the experiment was the worst performer overall, Touch + Tilt. However, we caution against drawing any conclusions regarding learning effects since the experiment consisted of only four trials per control mode. This is a potential topic for future research. Alternatively, improving tilt-based control modes may also help. For example, employing higher levels of tilt gain or reducing the required tilt range would require shorter, less drastic tilt motions and may yield higher performance with these conditions [20]. We note again that the maximum tilt range required in our experiment was only 30° in any axis, which was a comfortable range of control. Nevertheless, participants were sometimes observed (and our recorded tilt data confirms this) drastically tilting the tablet well beyond this range, even though further tilting would have no impact on their movement speed or orientation in the game. This drastic tilting might have influenced performance in a global sense, as it was observed across all tilt conditions (and hence it does not explain the inferior performance of Touch + Tilt). Using nonlinear tilt gain mappings (e.g., “tilt acceleration”), similar to the transfer functions used to handle pointer acceleration in modern desktop OS UIs, may also be worth consideration.

Based on our results, we offer design guidance to game developers considering the use of tilt control:

1. Tilt-based orientation hinders navigation: This was reflected in the difficulty participants had in terms of environment collisions and motion paths.
2. Tilt-based movement works well: The Tilt + Touch condition offered fairly good navigation performance. This is somewhat similar to marble maze games [7] – the player effectively rolls the character in the desired direction. *Tilt + Touch is therefore the best arrangement for exploring more complex control schemes (e.g., additional virtual face buttons).*
3. Tilt-based control is appropriate for aiming tasks: In games that do not include a navigation component, tilt control is a viable option for aiming. Tilt fared well in terms of accuracy and enemies destroyed.

Finally, we discuss some limitations of our study, and describe some opportunities for future work. First, we note that using Touch + Touch for a practice trial (i.e., the “tutorial” upon starting the experiment) effectively gave participants 5 trials with Touch + Touch (rather than 4 with the others) and may potentially create a slight bias in favour of the condition. However, we argue that as the most familiar condition we expect that any improvement due to practice would be minimal, since learning effects are more pronounced on unfamiliar conditions. The alternative of using one of our novel control modes would thus likely be more problematic, and not providing any tutorial trial at all would likely be even worse.

A second limitation is that the standard joystick configuration (i.e., Touch + Touch) uses two joysticks for three actions: movement, aiming, and shooting. Hence, we note that the distribution of those actions across the different control modes is imperfect. The issue is that a gyroscope cannot replicate the use of a discrete action event, such as a touch. Automating the player’s shooting, something rarely used in game designs, would have compromised the external validity of the study. Therefore, using a button to maintain the discrete event (shooting) was necessary in control modes employing tilt for aiming.

Joysticks are well-suited to independently control position and orientation. In contrast, it is considerably more difficult or impossible to use two gyroscopes for independent position and orientation control (with an additional button to fire). Thus, in future work, we plan to compare the Tilt-only control mode with an equivalent joystick-based control mode, i.e., one that controls both position and orientation simultaneously. This may isolate the impact of coupling vs. decoupling the movement and orientation controls.

Notably the synergy of tilt and touch neither significantly improved – nor hurt – performance. Specifically, Tilt + Touch offered average performance comparable to the standard dual-joystick control scheme, Touch + Touch. Our data did not reveal statistically significant differences between these conditions. Future experiments will look further at the effects of occlusion (e.g., player hands, virtual controls) and tilt combined with physical controls.

7. Conclusions

Our results suggest there is merit to the idea of tilt supporting touch control. While the tilt-augmented control modes tended not to outperform the standard Touch + Touch control mode, the Tilt + Touch control mode tended to offer comparable performance. It was also ranked a close second in terms of subjective preference – for many subjective preference questions, our data failed to detect a significant difference between Touch + Touch and Tilt + Touch, yet found significant differences between these two control modes

and the less preferred ones. Most notably, Touch + Tilt yielded significantly worse movement control, suggesting control problems when coupling touch-based movement control but tilt-based orientation.

This result was surprising, as Touch + Tilt used a virtual joystick for movement like Touch + Touch. It was thus expected to offer comparable movement performance, but potentially different aiming performance. This may reveal an inherent complexity in combining these two control schemes in this manner. Consequently, our results suggest that tilt may be best reserved for movement in games of this nature, with touch-based input used for aiming. Although we have offered several potential reasons for this result, ultimately, further testing is needed.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at <http://dx.doi.org/10.1016/j.entcom.2017.04.005>.

References

- [1] G. Alankuş, A.A. Eren, Enhancing gamepad FPS controls with tilt-driven sensitivity adjustment, in: International Conference on Computer Graphics, Animation and Gaming Technologies – EURASIA GRAPHICS 2014, Hacettepe University Press, 2014, Paper #13.
- [2] R. Balakrishnan, I.S. MacKenzie, Performance differences in the fingers, wrist, and forearm in computer input control, in: Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI '97, ACM, New York, 1997, pp. 303–310.
- [3] M. Baldauf, P. Fröhlich, F. Adegeye, S. Suetto, Investigating on-screen gamepad designs for smartphone-controlled video games, ACM Trans. Multimed. Comput. Commun. Appl. 12 (2015) 1–21.
- [4] S. Brewster, F. Chohan, L. Brown, Tactile feedback for mobile interactions, in: Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI 2007, ACM, New York, 2007, pp. 59–162.
- [5] K. Browne, C. Anand, An empirical evaluation of user interfaces for a mobile video game, J. Entertain. Comput. 3 (2012) 1–10.
- [6] K. Chu, C.Y. Wong, Mobile input devices for gaming experience, in: IEEE International Conference on User Science and Engineering – I-USER 2011, IEEE, New York, 2011, pp. 3–88.
- [7] C.I. Constantin, I.S. MacKenzie, Tilt-controlled mobile games: velocity-control vs. position-control, in: Proceedings of the IEEE Consumer Electronics Society Games, Entertainment, Media Conference – GEM 2014, IEEE, New York, 2014, pp. 4–30.
- [8] J. Cuaresma, I.S. MacKenzie, A comparison between tilt-input and facial tracking as input methods for mobile games, in: Proceedings of the IEEE Consumer Electronics Society Games, Entertainment, Media Conference – GEM 2014, IEEE, New York, 2014, pp. 0–76.
- [9] A.H. Cummings, The evolution of game controllers and control schemes and their effect on their games, in: The 17th Annual University of Southampton Multimedia Systems Conference, 2007.
- [10] T. Fritsch, B. Voigt, J. Schiller, Evaluation of input options on mobile gaming devices Technical Report, Freie Universität, Berlin, 2008.
- [11] P. Gilbertson, P. Coulton, F. Chehimi, T. Vajk, Using “tilt” as an interface to control “no-button” 3-D mobile games, Comput. Entertain. 6 (2008) 38.
- [12] Y. Guiard, Asymmetric division of labor in human skilled bimanual action: the kinematic chain as a model, J. Motor Beh. 19 (1987) 486–517.
- [13] A. Henrysson, J. Marshall, M. Billinghurst, Experiments in 3D interaction for mobile phone AR, Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia – GRAPHITE 2007, ACM, New York, 2007, pp. 187–194.
- [14] E. Hoggan, S.A. Brewster, J. Johnston, Investigating the effectiveness of tactile feedback for mobile touchscreens, in: Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI 2008, ACM, New York, 2008, pp. 573–1582.
- [15] T. Hynninen, First-person shooter controls on touchscreen devices: A heuristic evaluation of three games on the iPod touch, M.Sc. thesis, Department of Computer Sciences, University of Tampere, Tampere, Finland, 2012, 64 pages.
- [16] S. Järvelä, I. Ekman, J.M. Kivikangas, N. Ravaja, Digital games as experiment stimulus, in: Proceedings of 2012 DiGRA Nordic, Digital Games Research Association, 2012.

- [17] B. Lee, A. Oulasvirta, Modelling error rates in temporal pointing, in: Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI 2016, ACM, New York, 2016, pp. 857–1868.
- [18] S. Lee, S. Zhai, The performance of touch screen soft buttons, in: Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI 2009, ACM, New York, 2009, pp. 09–318.
- [19] L. Lew, T. Nguyen, S. Messing, S. Westwood, Of course I wouldn't do that in real life: advancing the arguments for increasing realism in HCI experiments, in: Extended Abstracts on Human Factors in Computing Systems – CHI EA 2011, ACM, New York, 2011, pp. 19–428.
- [20] I.S. MacKenzie, 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, ACM, New York, 2012, pp. 68–577.
- [21] R.P. McMahan, E.D. Ragan, A. Leal, R.J. Beaton, D.A. Bowman, Considerations for the use of commercial video games in controlled experiments, *Entertain. Comput.* 2 (2011) 3–9.
- [22] S. Medryk, I.S. MacKenzie, A comparison of accelerometer and touch-based input for mobile gaming, in: International Conference on Multimedia and Human-Computer Interaction – MHCI, International ASET, Ottawa, 2013, pp. 117.1–117.8.
- [23] M.R. Mine, P. Frederick, J. Brooks, C.H. Sequin, Moving objects in space: exploiting proprioception in virtual-environment interaction, in: Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques – SIGGRAPH '97, ACM, New York, 1997, pp. 9–26.
- [24] M. Oshita, H. Ishikawa, Gamepad vs. touchscreen: a comparison of action selection interfaces in computer games, in: Proceedings of the Workshop at SIGGRAPH Asia, ACM, New York, 2012, pp. 7–31.
- [25] H.H. Sad, F. Poirier, Evaluation and modeling of user performance for pointing and scrolling tasks on handheld devices using tilt sensor, in: Proceedings of the International Conference on Advances in Computer-Human Interactions – ACHI '09, IEEE, New York, 2009, pp. 295–300.
- [26] V. Sazawal, R. Want, G. Borriello, The Unigesture approach: one-handed text entry for small devices, in: Proceedings of Human Computer Interaction with Mobile Devices – MobileHCI 2002, Springer, Berlin, 2002, pp. 56–270.
- [27] R.J. Teather, I.S. MacKenzie, Comparing order of control for tilt and touch games, in: Proceedings of the Australasian Conference on Interactive Entertainment, ACM, New York, 2014.
- [28] R.J. Teather, I.S. MacKenzie, Position vs. velocity control for tilt-based interaction, in: Proceedings of Graphics Interface, CIPS, Toronto, 2014, pp. 51–58.
- [29] D. Wigdor, R. Balakrishnan, TiltText: using tilt for text input to mobile phones, in: Proceedings of the ACM Symposium on User Interface Software and Technology – UIST 2003, ACM, New York, 2003, pp. 1–90.
- [30] L. Zaman, I.S. MacKenzie, Evaluation of nano-stick, foam buttons, and other input methods for gameplay on touchscreen phones, in: International Conference on Multimedia and Human-Computer Interaction – MHCI, International ASET, Ottawa, Canada, 2013, pp. 69.1–69.8.
- [31] L. Zaman, D. Natapov, R.J. Teather, Touchscreens vs. traditional controllers in handheld gaming, in: Proceedings of the International Academic Conference on the Future of Game Design and Technology – FuturePlay, ACM, New York, 2010, pp. 183–190.
- [32] S. Zhai, P. Milgram, W. Buxton, The influence of muscle groups on performance of multiple degree-of-freedom input, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI '96, ACM, New York, 1996, pp. 08–315.