Movement Control Methods for Mobile Devices: An Empirical Study of Displacement Interfaces

Elias Bestard Lorigados(✉), I. Scott MacKenzie, and Melanie Baljko

York University, Toronto M3J 1P3, Canada
{elias39,mack,mb}@cse.yorku.ca

Abstract. We conducted a user study to compare four displacement interfaces on mobile devices. An Android application was developed with four displacement methods: Soft Joystick, up-down-right-left Buttons, Tilt Sensors, and Dual Soft Joystick. The study involved 12 participants using the four interfaces to control an object from a starting point to a goal point in five trials having different paths. The time (s), number of wall hits, efficiency (%), and out-of-path movement (%) per trial were logged to compare the methods. There was a significant effect of Displacement Method and Path on each dependent variable ($p < .005$). The Button method had 95.5%, 90.4%, and 84.2% less out-of-path movement than the Soft Joystick, Tilt Sensors, and Dual-Soft-Joystick methods, respectively. Also, the number of times the object missed the path using Buttons represented 10% of the total, followed by Dual-Soft-Joystick with 20% of the total. Tilt Sensors was 2.8% faster than Buttons (second fastest) in time per trial. Button reported the best numbers for efficiency with 58.8% on average per trial being 12% more efficient than Soft Joystick as the second best. In addition, there were statistical differences in post hoc tests ($p < .005$) between Button, the best overall method, and the remain interfaces on all dependent variables results. The participants indicated Buttons as the most comfortable method overall and Tilt Sensor as the best and most promising method.

Keywords: Displacement · Movement control · Joystick · Sensors · Buttons · Up-down-right-left Buttons · Mobile user interface

1 Introduction

Mobile devices are broadly used by all generations in daily life. The use of these gadgets varies depending on the needs, impediments, interests, and even preferences of each person. Individuals who suffer from certain physical or social impediments might use a cell phone, tablet, or any other specific interface to facilitate their interaction with the external world in order and exploit the device sensors for movement simulation. On the other hand, other individuals primarily use these devices for entertainment, such as filling leisure time with activities

X. Fang (Ed.): HCII 2022, LNCS 13334, pp. 3–21, 2022.
https://doi.org/10.1007/978-3-031-05637-6_1

like playing video games, where motion is present most of the time. Nowadays, almost every person commonly interacts with and uses these devices for a combination of the aforementioned purposes. Therefore, the study and development of new user interfaces have become a necessity.

If we examine specific interfaces, we can focus on the displacement or movement of an object from a starting point to a goal point. Examples include moving a cursor, virtual ball, or playing card, as well as controlling virtual racing cars, space ships, or other objects, including players, in games scenarios, such as shooting games, role-playing video games (RPG), sports games, etc. For the purpose of this research, we refer to "displacement" as the movement of an object from a starting point to another point. Displacement is commonly found in games, but we aim to target all applications that use displacement in some way or another.

This study focuses specifically on an empirical comparison of displacement from a starting point to a goal point in mobile devices. We aim to collect data using different game-like interfaces to control an on-screen object's motion and use that information for a comparison of the interfaces with a set of participants. We will examine which input method is better for movement control, which one is easier or more intuitive for users, and if these input methods can be improved. This comparison can help to enhance several areas of research and development of mobile user interfaces, particularly for games. The results will help to determine situations where one interface or another is preferable for users. Also, these data will facilitate the selection of a preferred interface for future mobile application development or game development.

1.1 Related Work

Plenty of work has been conducted in the area of displacement control interfaces. Research progress is particularly evident in the use of joysticks, arrow buttons, tilt sensors, and key pads on mobile computing devices for gaming. Examples of these improvements were studied by Chu and Wong in evaluating participant experience on hard and soft keypads [3]. They found that the input device directly affected the gaming experience, and offered a list of game genres best suited to various input modalities.

Constantin and MacKenzie performed a user study examining tilt-control to move an object in a maze [4]. They compared the control of a ball using velocity-control and position-control. They found position-control 16% faster, with similar accuracy compared to velocity-control. On the other hand, the participants preferred velocity-control.

Medryk and MacKenzie compared gaming performance during gameplay using tilt-control and touch-based input system [5]. They observed better results using touch input during gameplay. On the other hand, participants preferred tilt-control since it brought more challenges to the games. Similarly, Browne and Anand conducted a study evaluating mobile interfaces specifically in video games [2]. They performed a study with 36 participants using three interfaces, where they had a combination of accelerometer sensors, buttons, and a touchscreenbased interface. The reported that users preferred sensors over joysticks and buttons.

Other studies compared on-screen game-pad designs with sensor-controlled designs for mobile devices on video games. Baldau et al. conducted a comparison of four game-pad controllers using traditional input methods [1]. They specifically compared directional buttons, a directional pad, a floating joystick. And tilt control and observed that directional buttons were the most precise for direction-restricted navigation. Additionally, unintentional movements occurred while using the joystick and the use of tilt was rejected for some tasks.

In addition, Teather and MacKenzie evaluated the tilt-control method on mobile devices [6]. They investigated the performance of cursor control using devices with built-in accelerometers, different levels of tilt gain, and two selection modes. This study reported a tilt gain range of 50–100 to obtain optimal performance for movement time.

Similarly, Teather et al. proposed the use of tilt sensors coexisting with touch control in mobile devices [8]. They used two soft joysticks to independently control the moving and aiming in shooting games and included a standard dual analog control scheme. They also tested two options using tilt control in lieu of touch control for either movement or aiming, and a tilt-only control scheme. Conclusively, they report that touch-based control offered the best performance but tilt-based was a comparable interface.

Zhang et al. worked on an interface to control games based on different sensors [9]. Their results in the games studied indicate better performance using a joystick. However, participants still preferred using the sensors, which provided more freedom controlling the movement.

Input using a tilt sensor has also shown good results in comparison with touch-based interfaces in a study on motion sickness while gaming [7]. In the cited study, motion sickness was commonly seen with users who spend many hours gaming. This is an additional reason for testing the use of a tilt sensor in the present study. All the cited studies support velocity-control using a tilt sensor as an input method and stress the importance of studying the differences between movement interfaces. They inspired us to examine traditional and novel methods to control the movement of objects in mobile devices. Thus, the optimal values for tilt gain obtained by Teather and MacKenzie [6] were relevant to the development of our Android application. It was also found that directional buttons and the use of joysticks are common in the literature.

2 Method

A user study was conducted to compare four different movement interfaces in terms of performance differences and user preferences. Each participant completed five trials, each using a different path pattern, on the four different movement interfaces (see apparatus section for a description of the interfaces).

2.1 Participants

Twelve participants were selected to conduct the study and evaluation of the experiment. They were divided into four groups to counterbalance learning effects. Aiming to include variety, the study included daily smartphone users, avid gamers, and those who do not tend to play video games. Ages ranged from 18 to 35 years, with most of the participants between 21–30 years old. Six of the participants were male, six were female. The participants were selected from the local university campus and received $25 compensation for their assistance.

2.2 Apparatus

The study was conducted on an LG *G4* smart phone running the Android 8 operating system. The device had 5.5-in. display with a resolution of 1440 × 2560 pixels and 538 ppi density (Fig. 1).

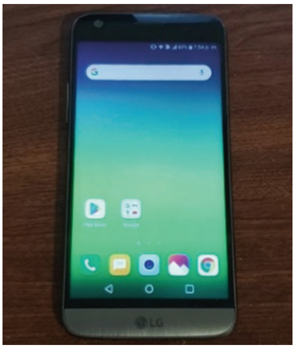


Fig. 1. Mobile device used in the user study, LG *G4*.

The software[1] was developed on Android Studio (4.1.1) using Java as the programming language for the main activities and XML for the graphical interfaces.

The application contains four interfaces for displacement control:

- Soft Joystick, Fig. 2(a)
- Dual Soft Joystick, Fig. 2(b)
- Up-down-right-left Buttons (Buttons), Fig. 2(c)
- Tilt Sensors.

Each interface controls the movement of an on-screen object differently. The Soft Joystick and Tilt Sensors interfaces give a 360° of movement control of the object while the Dual Soft Joystick and Buttons interfaces are limited to

[1] The software was developed using Demo_TiltBall as the starting point.

moving an object up, down, left, right, and diagonally. The Tilt Sensors interface uses the device's built-in tilt sensor and the tilt of the device to control the direction and speed of the object. Additionally, the Soft Joystick presents a wide circle containing a smaller circle in the center as shown in Fig. 2(a). This interface directs the object to where the smaller circle is positioned within the larger circle. The speed of object movement is computed by the distance of the smaller circle to the center of the joystick. Maximum speed is reached when the smaller circle touches the circumference of the joystick. The Buttons interface is maybe the most commonly seen interface for displacement control, see Fig. 2(c). It functions simply by moving the object in the direction of the pressed button. When pressing two buttons, such as right+up, movement is diagonal in the up-right direction. Finally, the Dual Soft Joystick interface is composed of two Soft Joysticks, as seen in Fig. 2(b). The left joystick moves the object up-down while the right joystick moves the object left-right. Similarly to the Buttons interface, when using both joysticks simultaneously, the object moves in a diagonal direction.

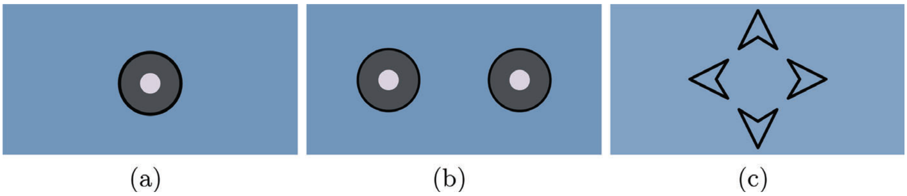


Fig. 2. Displacement interfaces: (a) Soft Joystick, (b) Dual Soft Joystick control, (c) Updown-right-left buttons.

The flow of the application is shown in Fig. 3. First, there was a setup state where the user chose the movement method to use during the trials. Users were asked to try the movement method using a test mode before starting the experiment. This allowed them to become familiar with the input methods. When the test ends the application returns to the setting state. This was followed by the experimental state, where the users completed five movement patterns or paths with each interface. Each path completed is consider a trial. See Fig. 4. The name of each path was given after its shape or directions they follow (N-North, S-South, E-East, W-West).

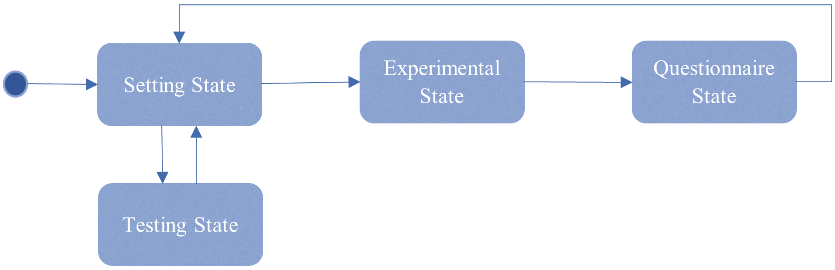


Fig. 3. Flow chart of the application showing how each state interacts with the next one.

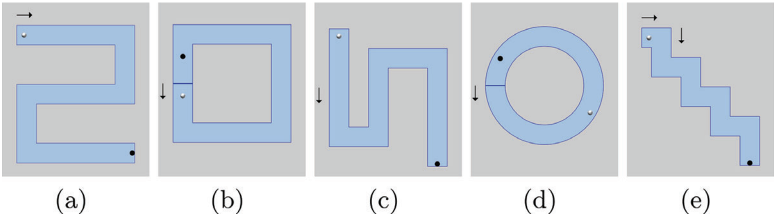


Fig. 4. Paths used in the user study: (a) ESWSE, (b) Square, (c) SENES, (d) Circle, (e) Stairs.

The participants then completed a questionnaire about the comfort of the displacement method after finishing with each input method. Lastly, data were collected, and the users were then directed to either the setting state to choose the next interface method to continue the experiment or to exit if they completed all interfaces. Each participant complete the five path trials with each displacement method.

The experiment interface was presented in a View panel where the participants could see and control an object – a virtual ball. There was a starting point (where the object originally starts), a path to follow, and a goal point. At the bottom, there was a panel with the input method for controlling the movement. Figure 5 shows a screenshot of the application layout during a demo run. The task involved controlling the object by moving it from the starting point to a goal point and returning to the starting point. The device vibrated each time the object touched the path borders while the object is inside the path. When the object completed the out-and-back movement, a sound was emitted and a dialog appeared indicating the beginning of the next trial, which started after the user pressed “OK”.

The timing of the trial began when the experiment state started and ended when the object touched the goal point on its way back. The logged data for each trial contained the method of displacement control, the time per trial, the optimal time for each path, the number of times the ball hit the path wall, the time the object was outside the path, the percentage of time the object was outside the path, and a set of coordinates tracking the ball movement. Notably,

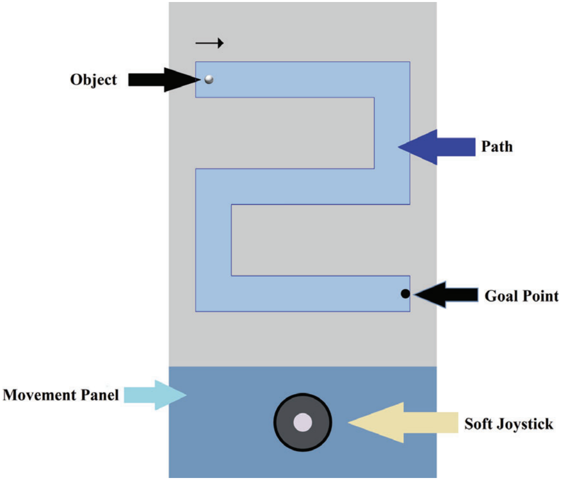


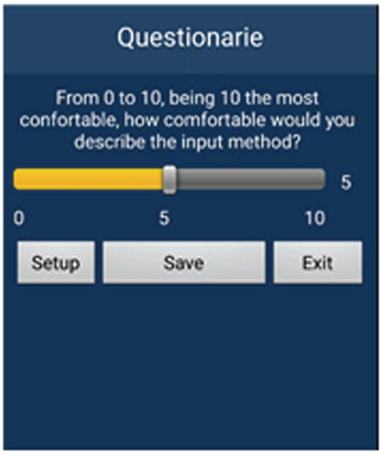
Fig. 5. Experiment state interface using Soft Joystick.

this percentage of time the object was outside the path was calculated by dividing the time outside the path by the time to complete the path multiplied by 100. The optimal path of each path is defined as the fastest time the object can complete each path following a line in the middle of the path.

After the five trials were completed with a chosen displacement method, the interface entered a questionnaire state where the participant indicated from 0 to 10 how comfortable they felt using the method, with 10 being the most comfortable (Fig. 6). The users saved their results and either went back to the setup state to continue the experiment with another displacement interface or exit the application. When they finished all interfaces, after closing the app, they answer a set of questions about the procedure.

2.3 Procedure

For the procedure, we ensured each participant understood the purpose of the experiment, what to do/expect, how long it would take for them to complete the experiment and which group they belonged to (see design section for a description of each participant group). They were asked to do one practice test with each interface so as feel comfortable with the displacement method. Participants were assigned to a group (for counterbalancing) and started the experiment with no breaks in between. When they finished the five trials on each input method, they saved the data by clicking a save button and continued with the next interface until they were done. After finishing all trials, participants indicated which method they liked the most and which one they felt would become the most practical after additional practice. The experiment took about 20 min per



Questionarie

From 0 to 10, being 10 the most confortable, how comfortable would you describe the input method?

Fig. 6. Questionnaire state.

participant to use the four input methods. We did not allow trial repetitions or incomplete experiments when performing the analysis. Figure 7 shows an example of a participant performing the experiment using the Dual Soft Joystick interface.

2.4 Design

The user study was a 4 × 5 within-subjects design with the following independent variables and levels:

- Displacement method: Soft Joystick, Dual Soft Joystick, up-down-right-left Buttons, Tilt Sensors
- Path: ESWSE, Square, SENES, Circle, Stairs.

In addition, we gathered subjective responses at the end with a questionnaire asking participants to indicate from 0 to 10 how comfortable they found each displacement method (higher scores better). Also, participants indicated their preferred method and the method they felt would work best after sufficient practice. The experiment took about 20 min per participant.

The dependent variables were time per trial (seconds), the number of wall hits per trial, the efficiency of each trial (%) defined in Eq. 1, and the percentage of out-of-path movement time for each trial (%).

$$Efficiency = \frac{OptimalTime}{Laptime} \times 100 \tag{1}$$

Participants were divided into four groups to counterbalance the testing order, thereby cancelling the learning effects using a balanced Latin square. The first group used the input methods in the following order: Tilt Sensor, Soft Joystick, Buttons, and Dual Soft Joystick. The second group used the order Buttons, Tilt Sensor, Dual Soft Joystick, and Soft Joystick. The third group used

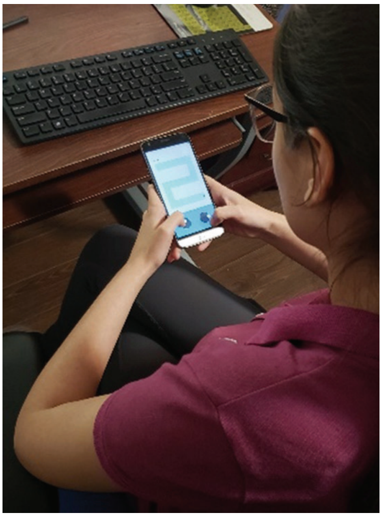


Fig. 7. A Participant performing the experiment using Dual Soft Joystick in path ESWSE.

the interfaces in the reverse order of the first group, while the last one used them in the reversed order of the second group. The total number of trials was 240 (= 12 participants × 4 displacement methods × 5 paths).

3 Result and Discussion

3.1 Time per Trial

The grand mean for time per trial was 26.0 s. Most trials (81 out of 240) took between 15.7 and 22.0 s to complete when 11.1 s was the optimal time average. Then, the effects of displacement method and path were statistically significant on the time per trial as expected, ($F_{3,24} = 5.78, p < .005$) and ($F_{4,32} = 20.57, p < .0001$) respectively. Additionally, the overall mean time for Tilt Sensor was 22.8 s having the fastest average of time per trial. The second fastest was reported by Buttons with an average of 23.50 s, followed by Soft Joystick with 26.3 s, and lastly Dual Soft Joysticks with 31.51 s. Thus, Tilt Sensor was 2.8% faster than Buttons and 27.5% faster than the slowest one. Figure 9 shows the mean time per trial (s) on each different displacement method.

It was observed that the time per trial on each path showed a similar behavior with all four displacement methods. Buttons and Tilt Sensors were the two fastest methods per path with the exception of the Circle path where the Buttons and Dual Soft Joystick were slowest. This result was interesting since both methods of movement control (Buttons and Dual Soft Joystick), as part of their intrinsic characteristics, can only control the object in a specific direction (right, left, up, down, and diagonals). Also, the circular shape of this path might affect the time participants take using those two movement interfaces. On the other hand, Tilt Sensors and Soft Joystick can direct the object in any direction from 0 to 360°, making them easier to control the object in a circular path Fig. 4(d). Figure 8 shows the time per trial (s) of each different movement control interface for each path.

A Bonferroni-Dunn post hoc test ($p < .05$) showed that Buttons-Dual Soft Joystick and Dual Soft Joystick-Tilt Sensor comparisons were the only significantly different pairs in terms of the time per trial. In addition, the times of ESWSE-Stair and Circle-Stair were the only two path pairs that differed significantly.

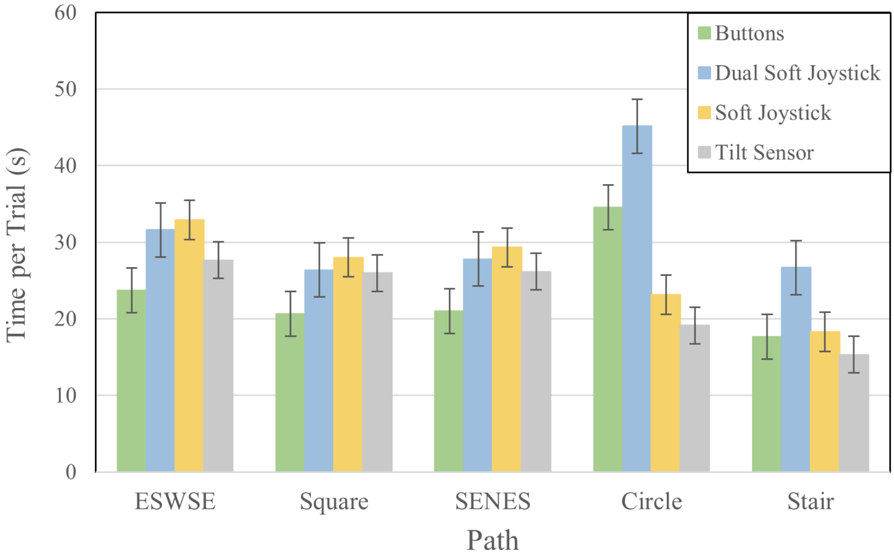


Fig. 8. Time to complete each trial (s) per displacement interface. Error bars show ±1 *SD*.

Figure 9 shows trace examples of participant trials. Figure 9(a) shows the fastest time of all trials made in 9.37 s by participant 8 in the Stair path using Soft Joystick. Figure 9(b) shows an average time of a trial by participant 8. It is interesting to observe an example of the use of Dual Soft Joystick on the Circle path. We can see that the control of the object become more complicated in circular shaped paths taking more time for the participant to complete the path and worsening the overall results. This example was completed in 23.9 s with 10 wall hits and 1.46% out-of-path movement. On the other hand, Fig. 9(c) shows the result of the same participant in the same path using Soft Joystick showing a better result of time, 14.3 s, 0% out-of-path movement, and smooth control of the object.

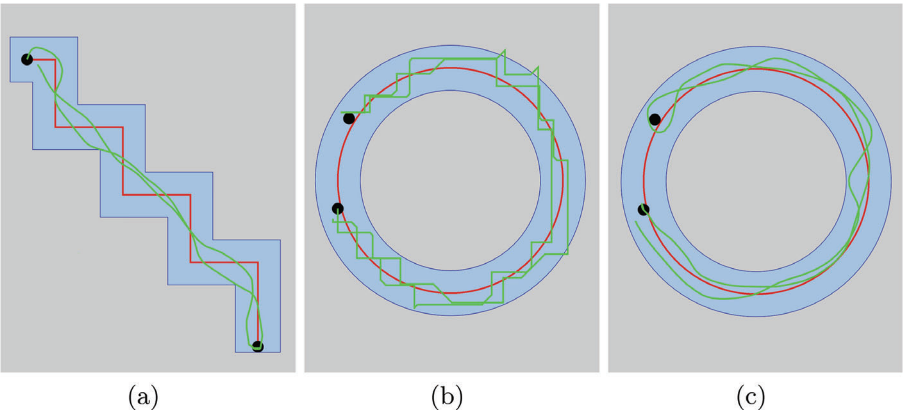


Fig. 9. Trace examples of a participant's trials. (a) Trace of the fastest trial, (b) Trace of trial in Circle path performed using Dual Soft Joystick, and (c) Trace showing a trial in Circle performed using Soft Joystick.

3.2 Number of Wall Hits

The study consisted of 240 trials with 40 trials per displacement method. The total number of times the object touched the wall (border of a path) was 701. Only 67 of this number happened with Buttons, representing 10% of the total. Dual Joystick reported 142 missed paths, which represents 20% of the total. Tilt Sensors and Soft Joystick reported 31% and 39% of the times the object touched the wall, 218 and 274 times respectively. See Fig. 10.

Number of Wall Hits per Trial. The grand mean number of wall hits per trial was 2.92 while most trials (93 out of 240) had between 0–2 wall hits. Analyzing the effects of the independent variables on number of wall hits, both (Displacement Method, Path) were statistically significant as expected ($F_{3,24} = 19.4, p < .0001$) and ($F_{4,32} = 7.52, p < .0005$) respectively. Buttons had the best performance, only touching the borders with a mean of 1.12 times per trial, and it is followed by Dual Soft Joystick with a mean of 2.37 times per trial. On the other hand, Tilt Sensors and Soft Joystick reported 3.63 and 4.57 wall hit times per trial. Thus, Buttons performed 75.5% better than Soft Joystick and 52.8% better than Dual Joystick as second best. See Fig. 11.

Over the five paths, it is interesting to point out that Buttons and Dual Soft Joystick performed similarly again in the number of missed paths per trial. Similarly, to time per trial, Buttons and Dual Soft Joysticks have a representative peak of missed paths on trial number four, which has a circular shape, as already mentioned. On the other hand, it is interesting how Tilt Sensor behaved over the trials in average. This method steadily decreased its number of wall hits per trial with a significant decrements in the Circle path and went up again as its usual values in the Stair path. Soft Joystick performed poorly when analysing each path separately. It decreased the wall hits for the middle paths but started

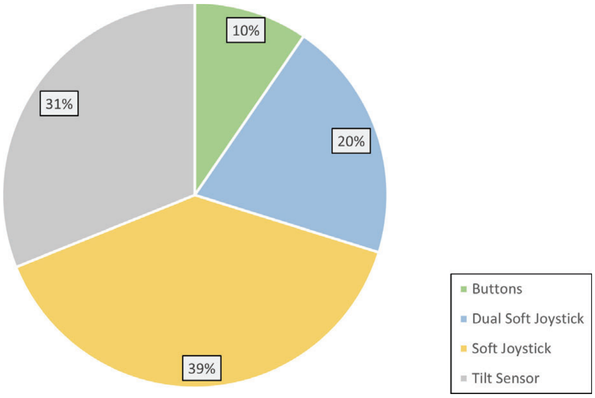


Fig. 10. The number of times the object missed the path as a percentage of the total.

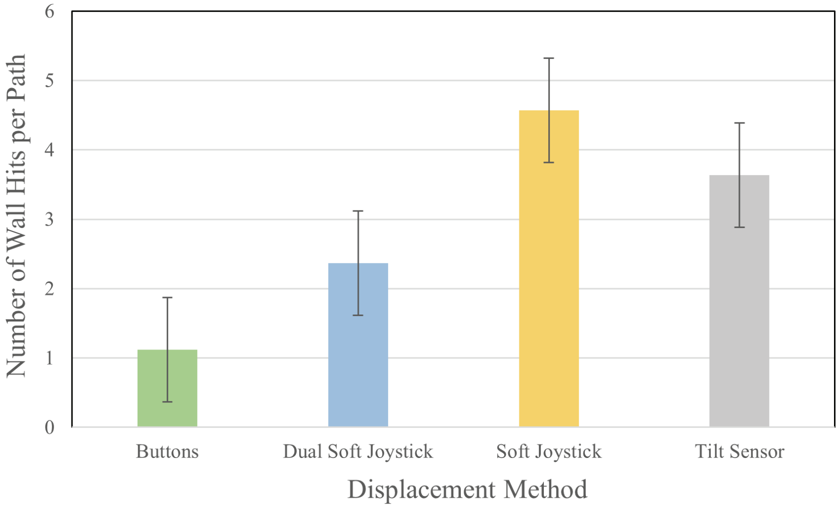


Fig. 11. Mean of the number of wall hits by displacement method. Error bars show ±1 *SE*.

to increase at Circle path. Overall, both, Buttons and Dual Soft Joystick, show improvements in the first three paths while Soft Joystick and Tilt Sensor perform better in the middle path and worst in the first and last path. See Fig. 12.

After conducting a Bonferroni-Dunn post hoc test ($p < .05$) it was shown that the pairs Buttons-Soft Joystick, Buttons-Tilt Sensor, and Dual Soft Joystick-Soft Joystick comparisons were significantly different. Additionally, the wall hits in the path pairs Square-Circle, Square-Stair, SENES-Circle, and SENES-Stair were significantly different.

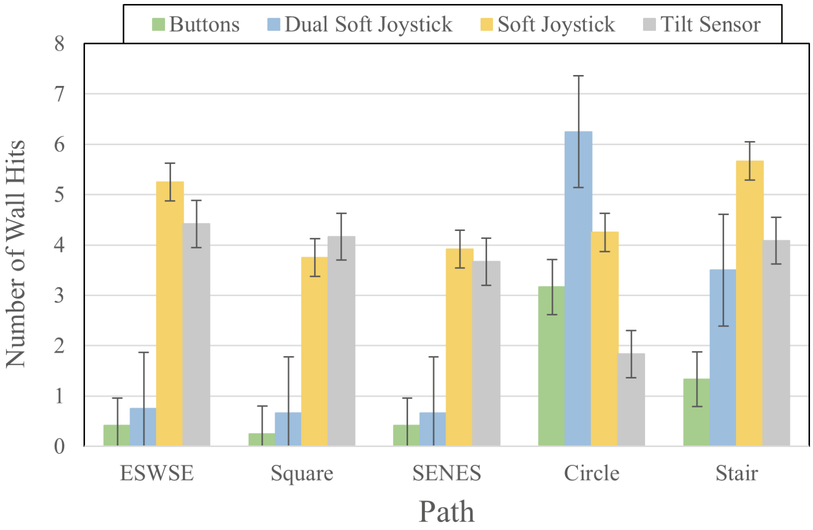


Fig. 12. The number of wall hits per trial by displacement method. Error bars show ±1 *SD*.

3.3 Efficiency

When analysing the efficiency of all displacement methods Tilt Sensor was removed because its results were not comparatively representative with the other methods. The results show extreme values when comparing the time participant took to complete the trials with respect to the possible optimal time for each path. Finally, most results of Tilt Sensor when analysing the efficiency are almost 0%.

The grand mean for efficiency per trial was 51.0% with most trials (63 out of 240) between 8% and 23% efficient. For both independent variables, displacement method and path, the effect on efficiency was statistically significant with ($F_{3,24} = 88.3, p < .0001$) and ($F_{4,32} = 53.2, p < .0001$), respectively. The average efficiency for Buttons was 58.8% which represents the best overall performance. The second one was reported by Soft Joystick with 52.5%, followed by Dual Soft Joystick with 43.9%. Thus, Buttons was 34% more effective than Dual Soft Joystick and 12% more effective than Soft Joystick.

Figure 13 shows trace examples of participant trials. Figure 13(a) shows an example of a trial with average efficiency performed by participant 4 using Soft Joystick and obtaining 48.3% of effectiveness with two wall hits. In addition, Fig. 13(b) shows the best efficiency obtained in the experiment by participant 9. It is interesting to highlight that this trial broke the estimate of optimal time of the Square path (16.5 s) using Buttons in 15.3 s with no wall hits – a perfect trial! Both had 0% out-of-path movement.

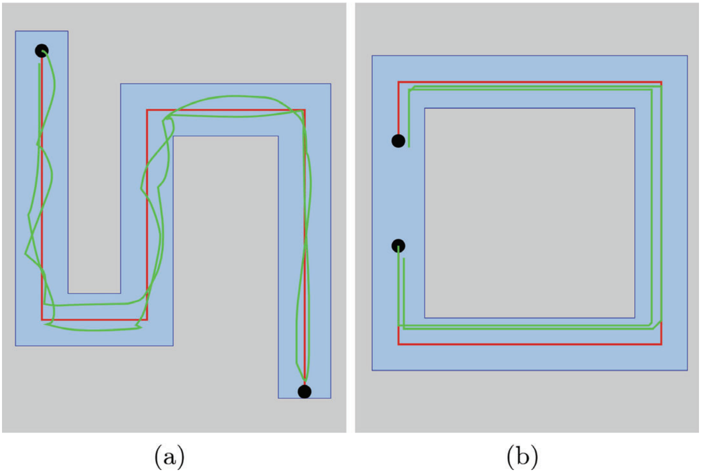


Fig. 13. Trace example of the participants' trials. (a) Example of an average efficiency trial, (b) Trace of the trial with best efficiency.

It was observed that the efficiency on all four different methods was decreasing overall. Soft Joystick seems to be more conservative about the effectiveness and stayed balanced around 50% on each different path. On the other hand, Dual Soft Joystick and Buttons showed similar results with their highest efficiency numbers in path number two and then a constant decrease with their lowest number in path number four. This also arguments that these two displacement methods do not perform well with circular shapes. See Fig. 14.

When analysing each pair of displacement method with respect to their efficiency conducting a Bonferroni-Dunn post hoc test it was found that the pairs Buttons-Dual Soft Joystick, Buttons-Tilt Sensor, Dual Soft Joystick-Tilt Sensor, and Soft Joystick-Tilt Sensor were significantly different between them. Similarly, the pair of paths ESWSE-Circle, Square-Circle, Square-Stair, and SENES-Circle were significantly different of each other. We can see how Circle path showed a difference with respect of most paths which is an interesting result to further study.

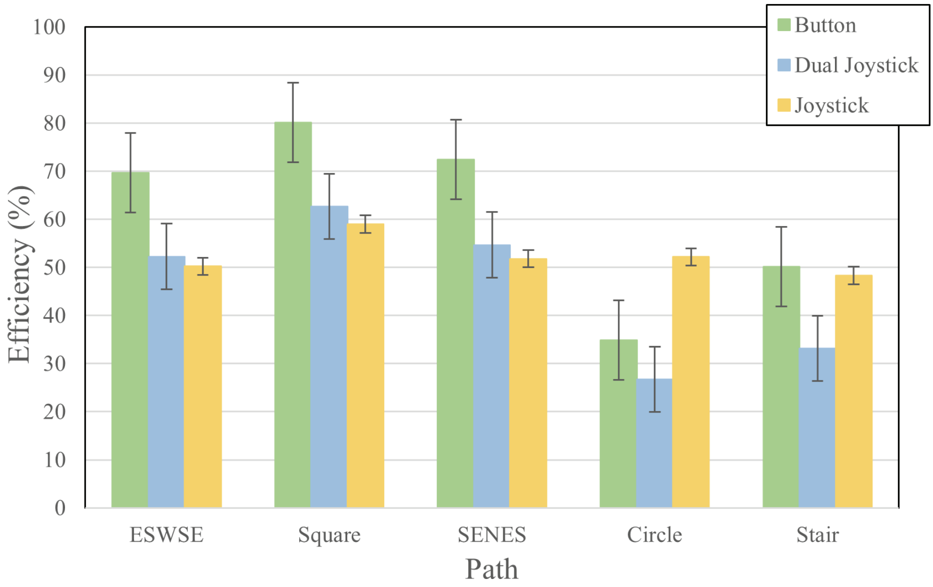


Fig. 14. The efficiency per trial by displacement method. Error bars show ±1 *SD*.

3.4 Percentage of Out-of-Path Movement Time

The grand mean of percentage of out-of-path movement time was 3.5% with the great majority (188 out of 240 trials) between 0% and 4.9% of out-of-path movement. The displacement method with best results in average was Buttons with only 0.4% of out-of-path movement while Soft Joystick reported an average of 7.9%. Dual Soft Joystick and Tilt Sensor presented similar results with 2.2% and 3.7%. Thus, Buttons was the most accurate overall being 95.5% more accurate than Soft Joystick (with the worst results) and 84.2% more accurate than Dual Soft Joystick. See Fig. 15.

There is a notary difference between Buttons and Dual Joystick over the five trials. After analyse the effects of displacement method and paths on the percentage of out-of-path movement, both were statistically significant ($F_{3,24} = 4.20, p < .05$) and ($F_{4,32} = 6.99, p < .0005$) respectively. The first method had an overall almost zero percent of out-of-path movement in all paths with zero in ESWSE and Stair. On the other hand, Soft Joystick oscillated between percentages above 10% and 3.7% as the lowest result in Square. The other two methods have a similar tendency over the five paths, they start decreasing and reported a spike in path number four and finished with a low percentage of out-of-path movement. It is worth highlighting that Buttons and Dual Soft Joystick had 0% average of out-of-path movement in two of the five paths for all trials. Figure 16 shows the overall percentages of out-of-path movement path on each trial.

When analysing each pair of displacement method with respect to their out-of-path movement conducting a Bonferroni-Dunn post hoc test it was found that only Buttons-Soft Joystick were significantly different form each other. Similarly, the pair of paths Square-Circle, Square-Stair, SENES-Circle, and SENES-Stair were significantly different of each other.

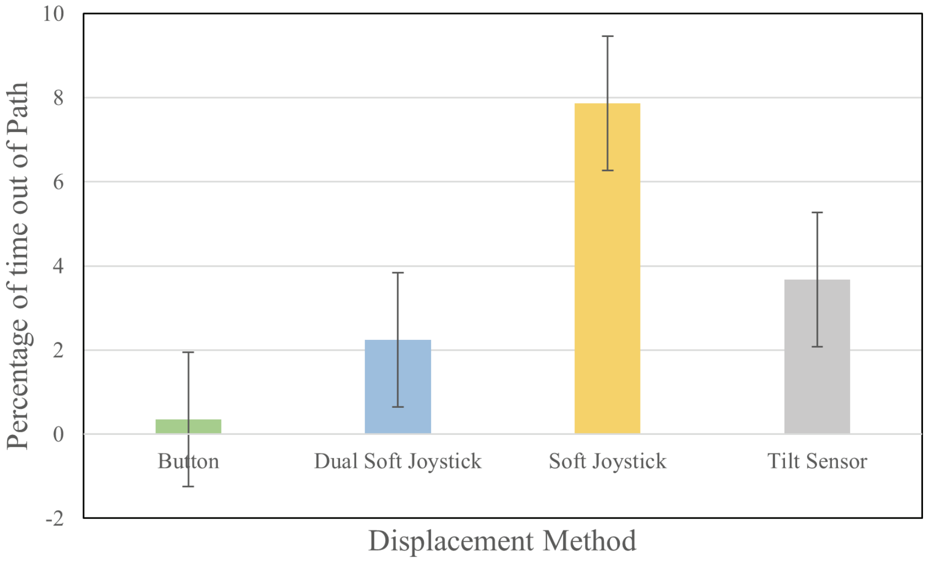


Fig. 15. Out-of-path movement (%) by displacement method. Error bars show ±1 *SE*.

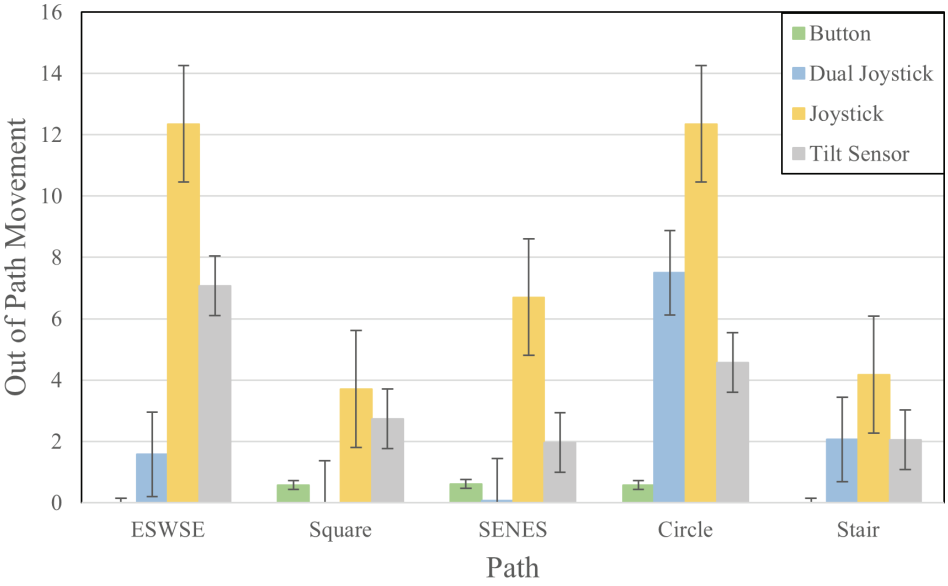


Fig. 16. Out-of-path movement (%) per trial by displacement method. Error bars show ±1 *SD*.

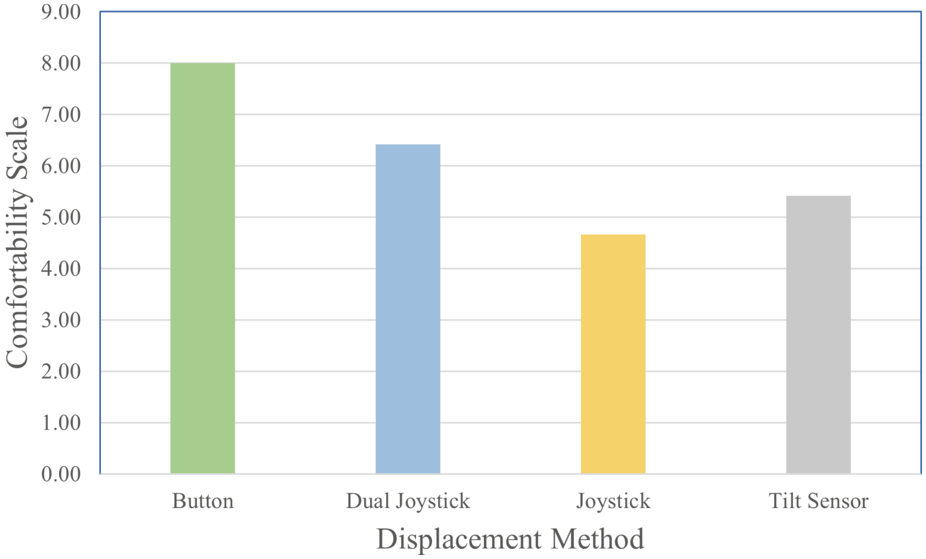


Fig. 17. Subjective comfort by displacement method.

3.5 Questionnaire

Each participant was asked to complete a questionnaire after they completed the five trials with each displacement method as well as they complete the experiment. The questionnaire asked how comfortable the participants felt from 0 to 10 with each method used right after they finished the five trials with each method. It also asked after the experiment ended which method they prefer and which method they felt will become better to use after time with practice. The most comfortable method was Buttons with a mean of 8.8 out of 10. It is followed by Dual Soft Joystick and Tilt Sensor with a mean of 6.4 and 5.42, respectively. Lastly, the least comfortable interface overall was Soft Joystick with a mean of 4.7 out of 10. See Fig. 17.

When analysing the preferences of each participant, Buttons and Tilt Sensor were the ones most preferable with five votes each. Then, Tilt Sensor was vote as the best method to used after time of practice followed by Buttons and Soft Joystick with three votes each.

4 Conclusions

A user study was conducted comparing four interfaces that control the movement of an object in mobile devices (Soft Joystick, Up-down-right-left Buttons, Tilt Sensors, and Dual Soft Joystick). The study revealed that Tilt Sensors (fastest time per trial method) was 2.8% faster than Buttons (the second fastest) in time per trial. On the other hand, the Buttons method had the lowest number of wall hits overall, outperforming Tilt Sensors as the third method with less wall hits.

Additionally, the Buttons method had the best efficiency overall. These results suggest that Buttons is a robust interface for movement control, having the best favourable results in number of wall hits, efficiency, and out-of-path movement while having the second-best results in time per trial.

Overall, participants gave a high comfort rating (>6) to Buttons and Dual Soft Joystick, thus confirming the high results for Buttons. On the other hand, the Buttons and Tilt Sensor were the most preferable even though Tilt Sensor had a low comfort rating overall. Additionally, the Tilt Sensor was also voted the best method to use after continued use. Even though users did not feel highly comfortable with the Tilt Sensor overall, its results and expectations with continued use highlight it as an strong method to exploit in games and movement control interfaces.

The effects of the independent variables (Displacement Method and Path) were statistically significant over all dependent variables while the groups for counterbalancing had no effect (as desired) in the statistical analyses. Overall, the Bonferroni-Dunn post hoc tests showed that the Buttons method is significantly different from the other interfaces over all variables. This reinforces that Buttons should be considered an irreplaceable option for displacement methods in different environments.

The study also revealed interesting patterns for the displacement interfaces. For example, Buttons and Dual Joystick performed similarly over all dependent variables in different path shapes while they showed significant differences in half of the dependent variables. This similarity is mostly seen as degraded performance in the Circle path while the Soft Joystick and Tilt Sensors performed consistently in all paths. These performance issues in the Circle path with respect to movement-control with the Buttons and Dual Soft Joystick raise questions regarding to their utility in different path shapes. Additionally, the post hoc analyses showed that the results with the Circle path are significantly different from the other paths in the study. This noteworthy observation could lead to future changes in evaluation strategies. Future work could examine participant performance on a greater variety of path shapes to determine which interface is better for each path shape. Furthermore, future studies should include additional methods of displacement control, more complicated scenarios, and consider a transition from 2D to 3D where a combination of different methods can bring new perspectives and opportunities to the game-development industry.

References

1. Baldauf, M., Fröhlich, P., Adegeye, F., Suette, S.: Investigating on-screen gamepad designs for smartphone-controlled video games. ACM Trans. Multimed. Comput. Commun. Appl. **12**(1s), 1–21 (2015)
2. Browne, K., Anand, C.: An empirical evaluation of user interfaces for a mobile video game. Entertainment Comput. **3**, 1–10 (2012). https://doi.org/10.1016/j.entcom.2011.06.001
3. Chu, K., Wong, C.Y.: Mobile input devices for gaming experience. In: 2011 International Conference on User Science and Engineering (i-USEr), pp. 83–88. IEEE, New York (2011). https://doi.org/10.1109/iUSEr.2011.6150542

4. Constantin, C.I., MacKenzie, I.S.: Tilt-controlled mobile games: velocity-control vs. position-control. In: 2014 IEEE Games Media Entertainment, pp. 1–7. IEEE, New York (2014). https://doi.org/10.1109/GEM.2014.7048091
5. Cuaresma, J., MacKenzie, I.S.: A comparison between tilt-input and facial tracking as input methods for mobile games. In: 2014 IEEE Games Media Entertainment, pp. 1–7. IEEE, New York (2014). https://doi.org/10.1109/GEM.2014.7048080
6. MacKenzie, I.S., Teather, R.J.: FittsTilt: the application of Fitts' law to tilt-based interaction. In: Proceedings of the 7th Nordic Conference on Human-Computer Interaction - NordiCHI 2012, p. 568. ACM, New York (2012). https://doi.org/10.1145/2399016.2399103
7. Stoffregen, T.A., Chen, Y.C., Koslucher, F.C.: Motion control, motion sickness, and the postural dynamics of mobile devices. Exp. Brain Res. **232**(4), 1389–1397 (2014)
8. Teather, R.J., Roth, A., MacKenzie, I.S.: Tilt-Touch synergy: input control for "dual-analog" style mobile games. Entertainment Comput. **21**, 33–43 (2017)
9. Zhang, D., Cai, Z., Chen, K., Nebel, B.: A game controller based on multiple sensors. In: Proceedings of the International Conference on Advances in Computer Entertainment Technology, ACE 2009, pp. 375–378. ACM, New York (2009). https://doi.org/10.1145/1690388.1690464