Touchscreens vs. Traditional Controllers in Handheld Gaming

Loutfouz Zaman Daniel Natapov Robert J. Teather

Department of Computer Science and Engineering
York University, Toronto, Canada
{zaman, dnatapov, rteather}@cse.yorku.ca

ABSTRACT

We present a study which compares touchscreen-based controls and physical controls for game input using Ubisoft's *Assassin's Creed: Altair's Chronicles.* Our study used the Apple *iPhone* as a representative touchscreen-based controller and the Nintendo *DS* for its physical control pad. Participants completed a game level four times on each platform. Level completion time and number of player deaths were recorded. Results indicate that physical buttons allowed significantly better performance than virtual buttons. Specifically, the number of character deaths on the *iPhone* was 150% higher than on the *DS*, while level completion time on the *DS* was 42% faster. The learning curve for the touchscreen version of the game was also steeper. Participants strongly preferred the physical buttons of the Nintendo *DS*. We conclude that either game designers should consider alternative input methods for touchscreen devices, or hardware designers should consider the inclusion of physical controls.

Categories and Subject Descriptors

H5.2. [User Interfaces]: Haptic I/O, Input devices and strategies (e.g., mouse, touchscreen), User-centered design

General Terms

Design, Experimentation, Human Factors.

Keywords

Tactile feedback, mobile device, touchscreen, mobile interaction, fingertip interaction, handheld gaming, haptic feedback, game input.

1. INTRODUCTION

Recently, there has been a departure from mobile devices using physical keypads and buttons towards devices exclusively using touchscreens [4]. This trend has been popularized by Apple's *iPhone* (see http://www.apple.com/iphone/). The versatile touchscreen minimizes the need for other physical input mechanisms while maximizing the display size relative to the device [8]. This allows for efficient pointing and touch-based input. By eliminating the need for keypads these devices allow for more compact form-factors [4]. Consequently, one can find a variety of such devices on the market today. Additionally, the recently released Apple *iPad* has the potential to popularize touchscreen-based handheld computing in the large screen mobile domain.

Gaming is arguably a primary use of these devices. Currently, there are far more games available for touchscreen handheld devices such as Apple's *iPhone* than for leading game-dedicated handhelds. In September 2009, during Apple's "It's rock and roll" event, the company announced that there are 21,178 game and entertainment applications available in the Appstore, as compared to 607 Sony *PSP* games and 3,680 Nintendo *DS* games [23]. As of February 2010, there are 4,118 game titles in the Android Market [1] and 254 game titles in Windows Marketplace for Mobile [18]. Additionally, the State of Game Development Survey revealed a recent surge of *iPhone* developers [11]. According to the survey, mobile developer support increased to 25% of game developers in 2009-2010, from 12% in 2008-2009. Nearly three quarters of these developers are targeting *iPhone* and *iPod Touch* development - more than twice the reported support for dedicated game platforms such as the Nintendo *DS* and Sony *PSP*.

Games for handheld devices often use the touchscreen and accelerometers for input. Game developers strive to utilize these input methods when developing games for these platforms. However, certain gameplay styles, such as "hack and slash", classic platformers, and first-person shooters are difficult to design without the use of physical-based controls. Consequently, Appstore games such as *Assassin's Creed: Altair's Chronicles*, *Resident Evil 4*, *Ghosts 'n Goblins*, *CastleGuard, N.O.V.A: Near Orbit Vanguard Alliance, Grand Theft Auto: Chinatown Wars*, etc. afford indirect input via virtual touchscreen-based controls. These games are designed with a standard physical controller in mind. Consequently, they display virtual controls on-screen, which the player interacts with via the touchscreen.

Popular opinion in the gaming community suggests that these touch-based directional pad controller "simulators" afford performance inferior to that of traditional physical button controllers. In this paper we investigate the validity of this claim using the game *Assassin's Creed: Altair's Chronicles.* We hypothesize that physical controls will indeed outperform touch-based controls, due to the haptic feedback they provide.

We also provide an overview of related work, addressing the limitations of hand-held touchscreen gaming. Mobile device manufacturers and game developers should use this work as a guideline, when developing games for these platforms.

ACM FuturePlay 2010, May 6-7, 2010, Vancouver, BC. Canada
Copyright 2010 ACM 978-1-4503-0235-7 ...$10.00

1.1 Related Work

1.1.1 Touchscreens

Touchscreens allow a great deal of flexibility, but there are a number of tradeoffs. Small touchscreens are impractical because the selection point is ambiguous due to the size and softness of the user's fingertip. This makes selection of small targets difficult and error-prone [4]. Even when screen sizes are large enough so to display larger targets (e.g., *iPhone*), finger-based interaction occludes parts of the display, covering up valuable pixels and making it harder to see the results of interaction [8]. Multi-touch displays exacerbate these problems [8]. Occlusion is reduced by stylus use, but this can be cumbersome as it requires the use of an additional object [8].

These issues may be addressed by *back-of-device* interaction – adding a touchscreen (or touch surface) on the back of the device, to avoid finger occlusion [4][13][22][25][26][27]. However, this kind of interaction has yet to be commercially deployed and accepted by the consumers. It is also unclear how it can be adopted for handheld gaming.

The Shift technique [24] redraws occluded pixels next to the fingertip. This is likely more useful in text-entry on soft keyboards, and may not be practical when simulating traditional game controllers on a touch-screen for hand-held gaming.

SideSight [8] mitigates the need for user input on the screen, or in fact on any part of the device itself. It instead uses proximity sensors to divert the user input region to the areas on either side of the device. While this solves the problem of occlusion on smaller screens it may be impractical for hand-held gaming as it requires the user to lay the device on a surface while using it.

1.1.2 Tactile Feedback

Tactile feedback is one possible solution to overcome gaming issues on touchscreen mobile devices. We consider research that compares keyboard text entry on mobile devices here. Hoggan *et al.* [14] demonstrated that physical keyboards and tactile touchscreens had significantly fewer typing errors than standard touchscreens. The authors suggest that better tactile actuators could improve performance of tactile touchscreens even further. The authors also suggest that device manufacturers should use tactile feedback in their touchscreen devices to regain some of the feeling lost when interacting on a touchscreen with a finger.

A similar study performed by Brewster *et al.* [7] demonstrated that vibrotactile feedback in touchscreen keyboards significantly improved text entry speed, reduced errors and improved error correction. Participant feedback strongly favored the tactile display.

4iThumbs is a screen overlay that adds minuscule bumps on the iPhone display, which correspond to the locations of the keys when typing in vertical mode [1][19]. This can be considered a low-cost solution. However, we know of no empirical research investigating the benefit of this approach.

SemFeel [28] is a tactile feedback technology for mobile touch-screen devices which provides the users with the semantic information about the object they are touching through multiple vibration motors embedded in the backside of the device. Studies have shown that users can distinguish ten vibration patterns in SemFeel, including linear patterns and clockwise circular patterns, at around 90% accuracy. The system supports more accurate interactions in an eyes-free setting than systems that offer no tactile feedback, or those that use a single vibration motor [28].

Pitts *et al.* [21] demonstrated that multi-modal feedback is preferred over visual feedback alone in a study involving driving simulation while using a handheld device. Combining visual, audible and haptic feedback was consistently rated highest. These results indicate the importance of haptic feedback in systems where user attention is divided and visual contact with the interface is limited. We suggest that these findings extend to hand-held gaming; the player's attention is focused on playing the game, and the touchscreen control buttons are occluded by the thumbs.

Some device manufacturers attempt to address the lack of tactile feedback. The Blackberry *Storm* [5], for instance, features SurePress technology in addition to the capacitive touch technology [3]. The manufacturer of SurePress claims that this provides better typing accuracy due to the addition of tactile feedback. With SurePress, the screen is essentially one physical button that the user can press down for a "click" action. The idea was first established by Mackenzie and Oniszczak [17], when they introduced a tactile touchpad. In their study, an actuator was placed beneath the touchpad. This allowed clicking and dragging on the surface in a more natural way than using an external button or doing a lift-and-tap operation (a series of three taps to activate click-and-drag mode). The authors' study found that the tactile condition was 20% faster than lift-and-tap and 46% faster than using a button for selection.

1.1.3 Game Design

Understanding controller and input issues is critical to game design, as the player interacts with the game by way of the inputs used. Yet, relatively little work has been done in this area. None of the above studies directly address issues in touchscreen hand-held gaming, such as occlusion, lack of tactile feedback and small screen size. It is possible to create new input paradigms for touchscreen devices. For example, one can use the accelerometer as input, and tilt the device to simulate turning a car steering wheel. If "traditional" (i.e., button-based) control methods are used, there are two alternatives: either develop games for devices with physical buttons, or emulate such controls on touchscreen devices. The latter is the current strategy adopted by game developers, who rarely have control over hardware design issues.

Some mobile device manufacturers, such as HTC, do deliver devices supporting both touch and type interfaces [15]. However, these devices are currently less popular than touchscreen only devices. As a result, game developers have little choice but to simulate physical controls on touchscreens.

When developing games for touchscreen devices, developers should consider the soft button sizes, because players normally use their thumbs for control. Parhi *et al.* recommend that target sizes should be at least 9.2 mm for single-target tasks and 9.6 mm for multi-target tasks for one-handed thumb use on touchscreen-based handhelds to avoid performance degradation [20].

Gamepad vibration is common in modern gaming consoles. It has been shown that predictive text entry on mobile devices is significantly faster for users with vibration alerts, raising speeds from 20 wpm to 23 wpm with practice [10]. Using vibration alerts in mobile games may similarly improve the gaming experience.

Brewster has shown that sonic enhancement of buttons could also improve performance [6]. There are some disadvantages to this approach. In particular, sounds could be intrusive and not suitable in noisy environments [7]. Nevertheless, button sounds should be considered whenever appropriate by game developers aiming to implement touchscreen-based controller simulations.

1.1.4 Other Hardware

Using trackballs for input in hand-held devices may improve mobile gaming. TouchBall [9] is a hand-held trackball-based touch/haptic interface that affords a great deal of flexibility in directional degrees of freedom. The device provides direct transfer of force feedback through frictional touch (with high sensitivity) and requires a relatively small amount of inertia. The device itself uses a compact hand-held design appropriate for mobile applications. The authors suggest that the device might be well-suited to sports games.

Finally, iControlPad [16] is a physical button controller cradle for the *iPhone* and *iPod Touch*. The phone or the *iPod* slides and locks in to place via the serial port allowing the user to use the externally connected buttons for control. This device has not yet been commercially deployed. There remain some major obstacles before this device becomes available. Currently, it only supports games for the Sony *Playstation*, *Nintendo Entertainment System*, *Super Nintendo* and *Gameboy Advance* emulators which require "jailbroken" devices. Console emulation typically violates copyright laws, and it is unlikely that this device will be approved by Apple. It is unknown whether this device is planned for Android phones. Another potential issue with this device is that it requires the user to carry the control pad along with the phone. It may also be inconvenient to answer incoming calls while using the device.

2. METHOD

The objective of our study was to compare touchscreen-based "virtual" controls to physical controls. We thus compared the in-game performance of playing a game on the Apple *iPhone 3G* to playing the same game on the Nintendo *DS Lite*.

2.1 Participants

Twelve unpaid participants (9 male) were recruited from the local community. The average participant age was 23.7 years (*SD* = 3.9). Nine participants were right-handed.

Participants completed a pre-experiment questionnaire asking demographic questions, as well as prior experience questions. The questions asked participants to rate their proficiency with touchscreen use, playing games on touchscreens, playing games on the Nintendo *DS* or the Sony *PSP*, and general game playing. The possible answers were:

- Beginner (rarely or never use/play)
- Intermediate (occasionally use/play)
- Advanced (frequently use/play)

Overall, eight participants were advanced game players, two were intermediate, and two were novices. In terms of *iPhone* usage, four participants were advanced, five were intermediate, and three were novices. In terms of *iPhone* gaming experience, only one participant was advanced, eight were intermediate, and three were novices. Finally, in terms of playing experience on the Nintendo *DS* or the Sony *PSP,* three participants were advanced, four were intermediate and five were novices.

2.2 Apparatus

The study used an Apple *iPhone 3G* for the touchscreen condition, and a Nintendo *DS Lite* for the physical button condition. The *iPhone 3G* and the *DS Lite* are both handheld systems, but they differ in a number of ways. Figure 1 depicts the *iPhone 3G*. It has a 9 cm LCD screen, and no physical controls. All input to the devices requires use of either the multi-touch capable touchscreen, or the built-in accelerometers. Figure 2 depicts a *DS Lite*. It has two 7.6 cm LCD screens, the bottom of which is a touchscreen. Input methods include the bottom touch screen, the directional pad (operated with the left thumb), the four action buttons (operated with the right thumb), and the two shoulder buttons (on the back, operated with index fingers).

Figure 1: Apple *iPhone 3G*. Image taken from http://www.apple.com/iphone/.

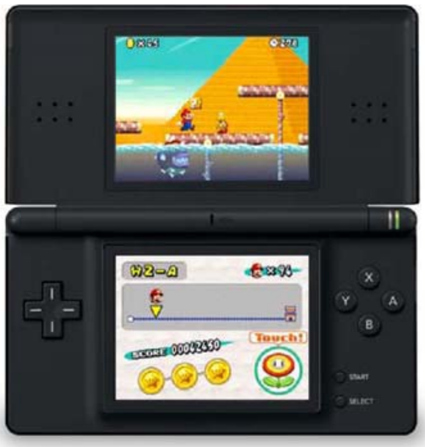

Figure 2: Nintendo *DS Lite.* Image taken from http://www.nintendodsi.com/.

To evaluate differences between physical and virtual buttons, we chose to use the game *Assassin's Creed: Altair's Chronicles* [12]. This game was chosen because it is available for both handhelds, and is largely the same on both, despite graphical differences. Other game titles that were investigated differ greatly between platforms. The game's main objective is primarily acrobatic action, such as climbing over obstacles, jumping over gaps, and walking on beams, mixed with stealth and combat. Most importantly, game controls are very similar between the two systems, with the *iPhone* version using on-screen widgets to emulate the *DS*'s physical controls. Note that one minor difference here is that the button layout was not identical, as the *DS* uses generic buttons to allow for many different games. The *iPhone*, on the other hand, allows custom controls for each game, due to the nature of virtual buttons.

The *DS* uses the directional pad to move the player character. The B button is used for jumping, and X and Y are used for a quick attack and a strong attack respectively. The shoulder buttons are occasionally used as well – for instance, the right shoulder button is held to maintain balance on narrow beams.

The *iPhone* version of the game simulates these physical controls with virtual buttons displayed on the touchscreen. A screenshot of the *iPhone* version of the game is shown in Figure 3. The blue circle at the bottom left of the screen simulates the *DS*'s directional pad, and is operated with the left thumb. Two attack buttons and a jump button are displayed at the bottom right. The key difference between the two systems is the lack of haptic feedback provided by the virtual touchscreen buttons of the *iPhone* version.

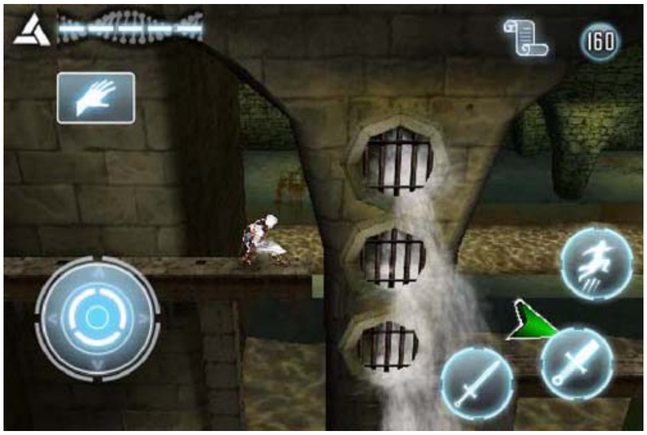

Figure 3: Screenshot of level 3-3 in *Assassin's Creed: Altair's Chronicles* on the *iPhone 3G.*

2.3 Procedure

We first explained the purpose of the experiment to participants. We then asked participants to give informed consent for their participation and to fill out the pre-experiment questionnaire.

The task involved completing one level of the game. The specific level used was level 3-3. We chose this level after careful consideration as we considered it to be the most linear level in the game. Unlike other levels, there is only one main possible route to take on this level. Additionally, the level has no distractions (e.g., conversations with NPCs, or mini-games) which would increase variability of the level completion task. A negative side effect of this choice is that level 3-3 is the 10th level in the game. Consequently, the learning curve was quite steep for participants, since they started playing in the "middle", and were not given a chance to play easier introductory levels. As a result, some participants had difficulty completing the task.

Participants were first given a practice trial to complete the level. During the practice trial, participants were given instructions, and were allowed to take as long as necessary to complete the level. After the practice trial, participants completed four recorded trials. Participants were asked to complete the level as quickly as possible while dying as few times as possible. We manually counted the number of times participants died during a trial, and timed how long it took to finish the level with a stopwatch.

When the player character dies, a loading screen appears, after which the player is returned to the last checkpoint. We stopped timing during the loading screen, and resumed timing once play resumed. Note that dying also affected level completion time. Upon resuming the game after dying, participants would have to redo parts of the level depending on how far back the last checkpoint was.

After the four recorded trials with the first device, participants switched to the other device, and were given another practice trial, followed by four recorded trials. On average, participants took an hour to complete the study, but a few novice players struggled with the task, and took longer.

Upon completing all trials with both devices, participants completed a post-experiment questionnaire to assess participant preference toward the devices.

2.4 Design

The experiment employed a 2 × 4 within-subjects design. The independent variables were:

- Device (*iPhone 3G*, Nintendo *DS Lite*)
- Trials (1 to 4)

Device order was counterbalanced between participants to avoid asymmetric learning effects between conditions. Trials lasted between roughly 3 minutes to as long as 15 minutes, depending on player proficiency and the number of deaths. Given that, participants performed 4 (Trials) × 2 (Device) × 12 (participants), a total of 96 trials were recorded. The dependent variables were number of deaths (count), level completion time (seconds), and modified level completion time (seconds, see below for details).

3. RESULTS AND DISCUSSION

3.1 Deaths

We first examine the number of player deaths. Player death could happen as a consequence of one of three events. The character could be killed in combat by enemies, be hit by a trap, or miss a jump and fall off a ledge. Traps in the chosen level are sewer water grates, which periodically spray water that pushes the character off the ledge if he is in front of one at the wrong time.

No participant actually died due to combat with enemies. This may be due to the relatively few enemies, as combat is not the main focus of the level. All the deaths that did occur were due to either falling, missing jumps, or the traps.

The number of deaths per trial with each input device can be seen in Figure 4 (standard error bars shown). Repeated measures analysis of variance revealed that there was a significant main effect for device ($F_{1,10} = 36.45, p < .0005$). The average number of character deaths when playing on the *iPhone* was 6.0, 150% higher than the *DS*'s 2.4. The order effect was not significant ($F_{1,10} = 1.17, p > .05$), indicating that counterbalancing was effective.

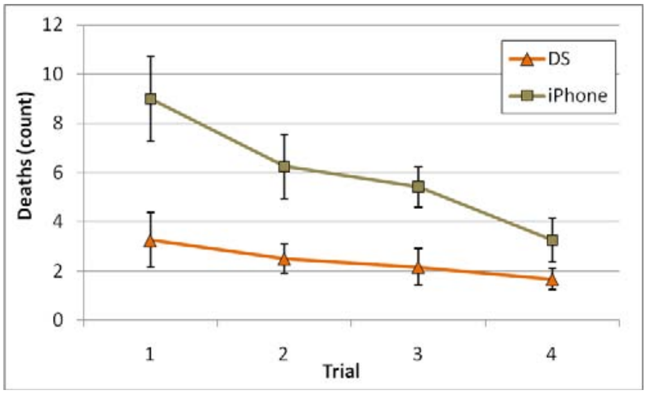

Figure 4: Number of deaths per trial. Error bars show ±1 *SE.*

It is clear that the device played a significant role in participants' ability to control the character, and prevent him from falling to his death. This confirms our hypothesis that the haptic feedback afforded by physical buttons results in superior control than the touchscreen.

As one would expect, participant performance also improved over time. The number of deaths decreased significantly on later trials for both devices. The main effect of trial was significant ($F_{3,30} = 6.43, p < .005$). In the *iPhone* condition, trial also had a significant effect on number of deaths ($F_{3,30} = 4.83, p < .01$). Tukey-Kramer multiple comparisons revealed that the average number of deaths in the fourth trial (3.25) was significantly lower from the average number of deaths in the first trial (9).

Learning rates for each device were also different. While the *iPhone* condition resulted in a far greater number of deaths in the early trials, the counts become closer towards the last trial. For the first trial, the average number of deaths with the *iPhone* was 9.0, 181% higher than the DS's 3.2. The difference is significant ($F_{1,10} = 9.86, p < .05$). Although these figures are much closer by the last trial, the number of deaths are still significantly different ($F_{1,10} = 9.97, p < .05$). The average number of deaths with the *iPhone* in the final trial was 3.3, still 94% higher than the *DS*'s 1.7. So while device performance is closer on the last trial, the number of deaths in the *iPhone* condition was still nearly twice that of the *DS* condition.

Despite this, it appears that while touchscreen-based virtual controls are difficult to use initially in this game, they improve with practice. It is possible that with further practice the results may get closer still. That said, we do not consider the results of our last trial competitive, and doubt that touchscreen performance would match physical control performance, even with additional practice.

3.2 Completion Time

We recorded level completion time to determine whether the benefits of haptic feedback improved the speed with which participants could complete the level. Ideally, completion time would be independent of the number of deaths, and instead only reflect how quickly participants were able to complete the level – regardless of how successfully. To that end, we stopped the timer during each death, and subsequent loading screen, and continued timing once the participant was able to continue playing.

Unfortunately, the game uses checkpoints, and upon dying and continuing, the player is transported back to the most recent checkpoint. In some cases, the checkpoint was close to the point where the character died, and dying had (nearly) no effect on completion time. However, if the character died a comparatively large distance away from a checkpoint, the death resulted in an increase in completion time, since the participant had to replay through part of the level. As a result, the number of deaths and level completion time are not completely separate measures, and dying did have a varying effect on completion time.

The results for level completion time are summarized in Figure 5 (standard error bars shown).

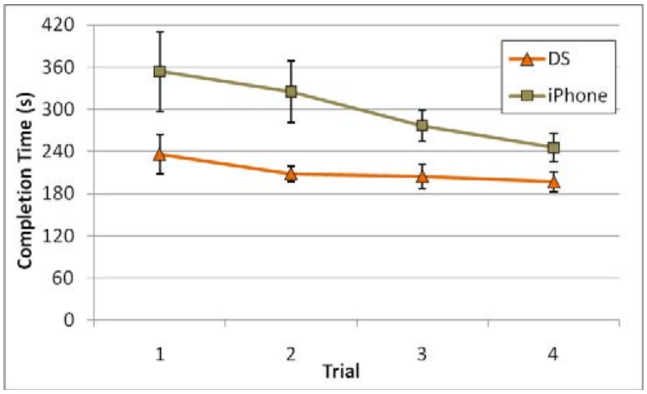

Figure 5: Completion time per trial. Error bars show ±1 *SE.*

The average level completion time using the *DS* was 211.4 seconds, 42.0% faster that the *iPhone*'s average level completion time of 300.3 seconds. There was a significant main effect for device ($F_{1,10} = 16.97, p < .005$) and trial ($F_{3,30} = 4.78$, $p < .01$). As with deaths, the group effect was not significant ($F_{1,10} = 0.11$, ns), indicating that counterbalancing was effective. A Tukey-Kramer analysis revealed that the average completion time in the first trial (295 seconds) was significantly greater than the average completion time in the last trial (221 seconds). We report further findings according to this grouping.

Much like the results for number of deaths, completion time with the *iPhone* improved faster than the *DS*. During the first trial, level completion time using the *DS* was 236.0 seconds, 50% faster than the iPhone's 354.0 seconds. The difference for the first trial is significant ($F_{1,10} = 5.90, p < .05$). The results on the last trial are closer, with the *DS*'s average completion time of 196.5 seconds only 24.8% faster than the *iPhone*'s 245.3 seconds. The difference for the last trial is still statistically significant ($F_{1,10} = 18.85, p < .005$), despite the gap between them narrowing.

The pattern for both completion time and number of deaths is very similar. The *iPhone* is severely worse for performance initially, but over time and with practice, it is possible to improve at a faster rate than improvement with the *DS*. Despite this, even with an hour of practice repeatedly performing the same task, on the last trial the *iPhone* was still worse by a significant margin both in terms of level completion time (19.9% slower), and the number of deaths (94% more deaths).

Our pre-experiment questionnaire showed that only one participant was an advanced *iPhone* game player. Users will generally improve faster with techniques with which they are less familiar, due to learning. This may be the reason for participants' performance improving more rapidly with the *iPhone*.

3.3 Modified Completion Time

As mentioned above, we intended for the number of deaths and level completion time to be independent of one another. However, due to the mechanic of checkpoints, a death would transport the character an indeterminate amount backwards in the level, thus increasing level completion time. To alleviate this, we attempted to average the time loss caused by dying, and to account for it in our calculation of level completion time.

The level had 9 checkpoints in total. In the best case, a death would occur right after a checkpoint, and incur a low backtracking cost of around one second. In the worst case, the character could make it nearly all the way to the next checkpoint, and a death would have a more expensive backtracking cost. To find the average worst case, we attempted to estimate the average playtime between two consecutive checkpoints. We recorded the time between every two consecutive checkpoints based on the playtime of an average performance participant, playing both on the *iPhone* and the *DS*. The results given below are averaged between the two systems.

The average worst-case time between two checkpoints was 22.6 seconds (SD 9.0). Assuming that on average, death frequency was somewhere between a worst case and a best case, we calculated a typical case death cost of 11.8 seconds as the average of the two.

Using this estimate of the average case death cost we calculated a modified completion time per trial as:

$$T_M = T - 11.8 \times D$$

Where T_M is the modified completion time, T is the measured completion time, and D is the number of deaths.

It is important to note that three assumptions are made for the calculation of modified completion time. First, we assumed that the participant we used to measure the time between checkpoints was representative. Second, we assumed that deaths occurred with an equal distribution between all checkpoints, which was unlikely, as some parts of the level are more difficult than others. Finally, we assumed that deaths occurred with an equal distribution about the middle of the worst and best case scenario, while the difficult parts of the level could be more heavily distributed towards the beginning or end of an interval between checkpoints. These assumptions may not be valid.

Consequently, we take modified level completion time as a general estimate at best, and not a precise value. That being said, the results for modified completion time are shown in Figure 6.

On average, the modified completion time was 183.1 seconds using the *DS*, 25.4% faster than the *iPhone*'s modified time of 229.7 seconds. The difference was statistically significant ($F_{2,11}$ = 8.66, p < .05). This suggests that even with the time loss due to deaths accounted for, playing was still slower with the *iPhone*, though not by as large a margin.

Overall, we feel confident stating that level completion time was slower with the *iPhone*, even without backtracking due to deaths. But due to the aforementioned broad assumptions, this result is not conclusive, and the 25.4% performance difference is certainly not accurate.

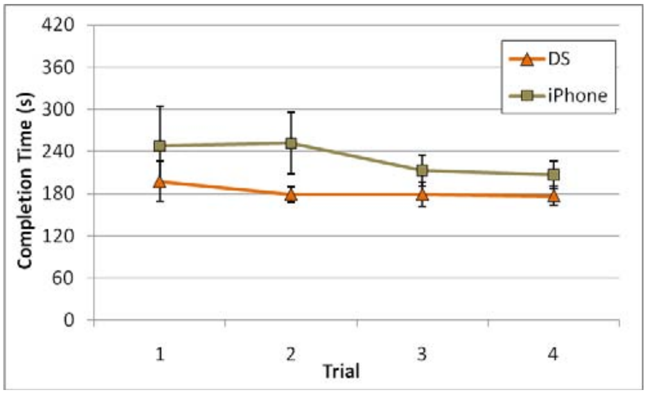

Figure 6: Modified level completion time per trial.

3.4 Qualitative Results

After the completion of the study, participants were asked to fill out a post-experiment questionnaire. The questionnaire asked which device participants preferred playing the game on. In terms of qualitative preference, all 12 participants selected the Nintendo *DS*. Indicative indeed!

4. Discussion

The nature of this experiment is complex and there are limitations. The screen sizes of the two devices are slightly different. The layout of the buttons on the right side is different for the two devices. On the *DS* the dedicated R button is used to toggle walking instead of running. On the *iPhone* sliding the circle on the directional pad widget all the way in one of the four directions, causes the character to run. On the other hand, displacing the circle slightly causes the character to walk. Hence, the direction control on the *iPhone* is more akin to a virtual analog stick than a direction pad. Rolling is activated automatically in the *iPhone* version of the game whenever a low-rise obstacle is traversed. Similarly to walking, the dedicated L button is used for rolling on the *DS*. Overall though, these differences between the devices are relatively minor, and would not account for the large performance differences we observed. Moreover, most of these differences should actually *favour* the *iPhone* condition, as they reduce the cognitive load on the player, making the game slightly easier on that platform. Therefore, we are confident that it was the haptic feedback which caused the performance improvement.

As another complication, the checkpoint issue makes it difficult to directly compare the completion time for each device. While one may think that number of deaths can be correlated to completion time, in fact only 65% of the data can be explained by the linear correlation (R^2=0.65). This is not surprising, as the time it takes to get from the last check point to the place of death will not be the same for all deaths and all checkpoints. When the number of deaths becomes high, it becomes particularly difficult to accept completion time as a valid measure. Thus, we believe that for this particular game the number of deaths is a more accurate estimate of controller performance.

We attempted to minimize differences and increase uniformity in participant performance, to remove variability. For instance, participants were instructed to kill all encountered enemies encountered before proceeding. Otherwise it was possible to avoid enemy confrontation and potentially complete the level faster. Additionally, level 3-3 features a closing gate where the character has to perform a sequence of acrobatic actions in a timely manner before the gate closes (Figure 7). As the gate starts closing, it is possible to roll under it on the Nintendo *DS* version (by holding the L button), but not on the *iPhone* version as it does not have a control button for rolling. We instructed the participants that they were not allowed to use the rolling feature with the gate while playing the level on the *DS* to match the limitation of the *iPhone* version.

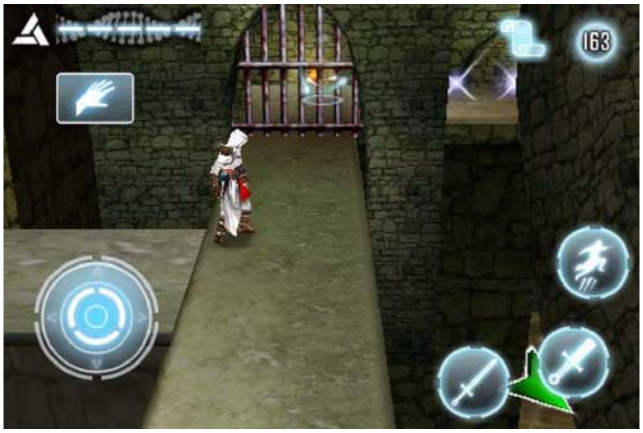

Figure 7: Screen shot of the closing gate sequence in level 3-3 of *Assassin's Creed: Altair's Chronicles* on the *iPhone 3G.*

Overall, though we tested only one game, we believe our results to be (in general) representative of the difference between touchscreen input and physical control input. The results are best taken as design guidelines, which suggest certain measures be implemented for touchscreen-based handheld devices to address the limitations of the hardware for gaming.

5. Conclusion

Previous studies have confirmed that touchscreen text entry and other types on interaction in handheld devices are less efficient than the devices that afford tactile feedback. The purpose of our study was to investigate whether these findings can be generalized to the interaction involved in handheld gaming. We confirmed these findings using quantitative and qualitative methods for level 3-3 of *Assassin's Creed: Altair's Chronicles*. This game was chosen because it uses a directional pad for control and is available for both Nintendo *DS* and the *iPhone*. It turned out that touchscreens are much worse in gameplay performance initially, and while performance improves with practice, physical controls still offer a significant performance benefit, even after an hour of practice.

We conclude that game designers should devise alternative control mechanisms that play to the strengths of the device, rather than trying to emulate controls that simply do not work on touchscreens. Alternatively, to improve their versatility as gaming devices, touchscreen handhelds could either include physical controls or some other form of tactile feedback, as discussed earlier in Sections 1.1.2 and 1.1.3.

6. Future Work

Due to the complex nature of our experiment, generalizing these findings for all types of games that use emulated physical controls may be a big step. Nevertheless, we believe that our results generated enough interest for future work.

One of the directions for future research could be to investigate which kinds of gameplay are affected the most by the lack of tactile feedback. Another possible area of future research can be to investigate the internal validity of hard vs. soft directional pad controllers with Fitts' tasks in better controlled experiments.

Back-of-device interaction can be investigated for hand-held gaming as a possible alternative for directional pad controls. At the moment, it is not clear if *back-of-device* interaction is transferrable to handheld gaming.

Touchscreen tactile feedback in handheld devices has been demonstrated as beneficial. In the future, we will investigate whether these findings are transferrable to handheld gaming. We will investigate if device vibration has an effect on soft directional pads in handheld gaming.

Some Android devices, such as the Nexus One, use vibration feedback during text entry or when clicking on GUI elements. Moreover, a Super Nintendo emulator app called *SNesoid* from the Android Market features the use of vibration feedback for its soft directional pad. In the future we plan to implement our own soft directional pad for the Nexus One and use a Fitts' task to test the benefit of vibration feedback.

We are also planning to investigate if a 4iThumbs overlay is effective for text entry. If proven effective we will create our own version of the overlay for gaming and investigate its effectiveness.

7. REFERENCES

[1] 4iThumbs, 2010, http://www.4ithumbs.com/, Access date: March 13, 2010

[2] Androlib, 2009, http://www.androlib.com, Access date: February 17, 2010

[3] Arthur K., 2008, The BlackBerry Storm's Three-State Touchscreen, http://www.touchusability.com/2008/10/the-blackberry-storms-three-state-touchscreen.html, Access date: March 4, 2010

[4] Baudisch, P. and Chu, G. 2009. Back-of-device interaction allows creating very small touch devices. In *Proceedings of the 27th international Conference on Human Factors in Computing Systems* (Boston, MA, USA, April 04 - 09, 2009). CHI '09. ACM, New York, NY, 1923-1932.

[5] Blackberry, 2010, http://www.blackberry.com/blackberrystorm/, Access date: March 4, 2010

[6] Brewster, S. 2002. Overcoming the Lack of Screen Space on Mobile Computers. *Personal Ubiquitous Comput.* 6, 3 (Jan. 2002), 188-205.

[7] Brewster, S., Chohan, F., and Brown, L. 2007. Tactile feedback for mobile interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, 159-162.

[8] Butler, A., Izadi, S., and Hodges, S. 2008. SideSight: multi-"touch" interaction around small devices. In *Proceedings of the 21st Annual ACM Symposium on User interface Software*

and Technology (Monterey, CA, USA, October 19 - 22, 2008). UIST '08. ACM, New York, NY, 201-204.

[9] Choi, M. and Kim, G. J. 2009. TouchBall: a design and evaluation of a hand-held trackball based touch-haptic interface. In *Proceedings of the 27th international Conference on Human Factors in Computing Systems* (Boston, MA, USA, April 04 - 09, 2009). CHI '09. ACM, New York, NY, 1535-1538.

[10] Dunlop, M. D. and Taylor, F. 2009. Tactile feedback for predictive text entry. In *Proceedings of the 27th international Conference on Human Factors in Computing Systems* (Boston, MA, USA, April 04 - 09, 2009). CHI '09. ACM, New York, NY, 2257-2260.

[11] Gamasutra, 2010, State Of Game Development Survey Reveals iPhone Support Surge, Wii Lull, http://www.gamasutra.com/view/news/26846/State_Of_Game_Development_Survey_Reveals_iPhone_Support_Surge_Wii_Lull.php, Access date: Ferbruary 9, 2010

[12] Gameloft, 2008, Assassin's Creed: Altair's Chronicles, 2010, http://en.gameloft.ca/iphone/assassins-creed/, Access date: February 9, 2010

[13] Hiraoka, S., Miyamoto, I., Tomimatsu, K. Behind Touch, a Text Input Method for Mobile Phones by The Back and Tactile Sense Interface. *Information Processing Society of Japan, Interaction 2003*. p. 131-138.

[14] Hoggan, E., Brewster, S. A., and Johnston, J. 2008. Investigating the effectiveness of tactile feedback for mobile touchscreens. In *Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 1573-1582.

[15] HTC, 2010, HTC Products, http://www.htc.com/products, Access date: February 18, 2010

[16] iControlPad 2010, http://www.icontrolpad.com/, Access date: February 20, 2010

[17] MacKenzie, I. S. and Oniszczak, A. 1998. A comparison of three selection techniques for touchpads. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Los Angeles, California, United States, April 18 - 23, 1998). C. Karat, A. Lund, J. Coutaz, and J. Karat, Eds. Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co., New York, NY, 336-343.

[18] Microsoft 2010, Windows Marketplace for Mobile, http://marketplace.windowsphone.com, Access date: February 17, 2010

[19] Murph D., 2009, 4iThumbs overlay adds a tactile keyboard to your iPhone, Engadget, http://www.engadget.com/2009/11/25/4ithumbs-overlay-adds-a-tactile-keyboard-to-your-iphone-sorta/, Access date: March 13, 2010

[20] Parhi, P., Karlson, A. K., and Bederson, B. B. 2006. Target size study for one-handed thumb use on small touchscreen devices. In *Proceedings of the 8th Conference on Human-Computer interaction with Mobile Devices and Services* (Helsinki, Finland, September 12 - 15, 2006). MobileHCI '06, vol. 159. ACM, New York, NY, 203-210.

[21] Pitts, M. J., Williams, M. A., Wellings, T., and Attridge, A. 2009. Assessing subjective response to haptic feedback in automotive touchscreens. In *Proceedings of the 1st international Conference on Automotive User interfaces and interactive Vehicular Applications* (Essen, Germany, September 21 - 22, 2009). AutomotiveUI '09. ACM, New York, NY, 11-18.

[22] Sugimoto, M. Hiroki, K. (2006). HybridTouch: an intuitive manipulation technique for PDAs using their front and rear surfaces In *Proceedings of the 8th Conference on Human-Computer interaction with Mobile Devices and Services* (Helsinki, Finland, September 12 - 15, 2006). MobileHCI '06, vol. 159. ACM, New York, NY, 137-140.. In *Proc.* MobileHCI '06, p. 137-140.

[23] Topolsky, J., 2009, Live from Apple's 'It's only rock and roll' event, Engadget, http://www.engadget.com/2009/09/09/live-from-apples-its-only-rock-and-roll-event/, Access date: February 6, 2010

[24] Vogel, D. and Baudisch, P., 2007. Shift: A Technique for Op-erating Pen-Based Interfaces Using Touch, In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, 657-666.In *CHI '07,* pp. 657–666.

[25] Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C. 2007. Lucid touch: a see-through mobile device. In *Proceedings of the 20th Annual ACM Symposium on User interface Software and Technology* (Newport, Rhode Island, USA, October 07 - 10, 2007). UIST '07. ACM, New York, NY, 269-278.

[26] Wigdor, D., Leigh, D., Forlines, C., Shipman, S., Barnwell, J., Balakrishnan, R., Shen, C. 2006. Under the table interaction. In *Proceedings of the 19th Annual ACM Symposium on User interface Software and Technology* (Montreux, Switzerland, October 15 - 18, 2006). UIST '06. ACM, New York, NY, 259-268. Under the Table Interaction. *In Proc. UIST'06,* 259-268.

[27] Wobbrock, J. O., Myers, B. A., and Aung, H. H. 2008. The performance of hand postures in front- and back-of-device interaction for mobile computing. *Int. J. Hum.-Comput. Stud.* 66, 12 (Dec. 2008), 857-875

[28] Yatani, K. and Truong, K. N. 2009. SemFeel: a user interface with semantic tactile feedback for mobile touch-screen devices. In *Proceedings of the 22nd Annual ACM Symposium on User interface Software and Technology* (Victoria, BC, Canada, October 04 - 07, 2009). UIST '09. ACM, New York, NY, 111-120.