Apple's 3D Touch Technology and its Impact on User Experience

Nicolas Suarez-Canton Trueba

March 18, 2017

Contents

List of Figures

List of Tables

Abstract

Approximately 2 years ago, Apple Inc. introduced the first of its lineup of devices to include 3D Touch, the iPhone 6S. This touchscreen technology is responsive to the amount of pressure that users apply on the screen, and thus developers can use it to add contextual functions to the present-on-screen software. This study presents three different experiments that test 3D Touch as: (a) a input device to accurately set a parameter, (b) a less disruptive way to interact with notifications and (c) an alternative error recovery mechanism. Although only experiment (a) was fully implemented and tested, there does not seem to be an improvement of user experience. This experiment was a task in which users were asked to input a rating using 3D-Touch or a traditional slider. The users performed better in terms of number of attempts and time taken on the slider interface; however, they appeared substantially more engaged when interacting with the 3D-Touch interface. Experiments (b) and (c) remain as future work, and the user engagements suggest that 3D-Touch could be better suited for its implementation within a video-game.

1 Introduction

On September 25, 2015 Apple Inc. released the latest version of its flagship smart-phone. This improved device, the iPhone 6S, implemented the company's newest touchscreen technology: 3D Touch. This technology allows users to interact with an iPhone application based on how hard they press the display. From a user experience standpoint, the introduction of 3D Touch is revolutionary. Software developers working on the iOS platform can now make use of this technology to create applications and games that are responsive to how deeply users press the display. For instance, a user can now press on items within an app's view to see previews of additional content. 3D Touch is ideal for creative applications. For example, Autodesk improved Sketchbook–a digital painting mobile application–so it includes pressure-sensitive brushstrokes on 3D Touch devices. Apple's first-generation iPhone brought multi-touch screen technology to the consumer market, and revolutionized consumer expectations in terms of user interface for hand-held devices [2]. The goal of this research project is to develop a series iOS application with 3D Touch capabilities in order to establish whether it has a positive impact on user experience. This project aims to evaluate the effect of 3D touch on a mobile application by comparing its performance to that of a traditional "2d" interface. This entails the implementation of two versions of each application: (a) one that uses 3D Touch to improve the user experience, and (b) one that runs on older devices and aims to simulate the experience without pressure sensitivity. The following sections present three different scenarios which may show that 3D Touch increases user performance. These experiments single out 3D Touch in such a way that users are forced to use it to complete each of the designated tasks. Further, each task provides a clear metric which will be use to quantify user performance, not preference in regards to either multi-touch or 3D touch interfaces.

2 Project Objectives

The objective of this project is to analyze the impact of 3D Touch on the user experience of iOS devices. Indeed, this project requires a 3D Touch enabled device. In this case, an iPhone 6s will be used as the testing device. Further, assessing the performance of 3D touch involves an interface that has been designed with the aim of testing particular features of 3D Touch. This study will test the following: (a) an assessment of 3D-Touch's accuracy, (b) 3D Touch notifications and cognitive overhead, and (c) 3D Touch as an error recovery mechanism. The following section provides a breakdown of how each experiment will be conducted and what metrics will be obtained.

3 Experiment Design

3.1 Assessment of 3D-Touch's Accuracy

The first experiment aims to determine whether 3D-Touch could be used as an input mechanism. In mobile applications, it is common to find screens that require the user to enter a numeric value within a range. For instance, user are often asked to rate their experience using a mobile application in a range from 0 to 5. Most implementations accomplish this task by using a slider: as the user moves the slider farther away from the starting value, the value that would be set as the rating increases. When the user releases his finger from the screen, the last value that was displayed on the slider prevails as the input value for the rating prompt. This experiment translates the idea of using a slider to set a parameter into the domain of 3D-Touch.

Before describing the experiment, it is necessary to address previous work on the field of "3D" interfaces to understand the design decisions. Previous work by Ramos et al. showed that appropriate visual feedback was a critical factor of user performance. They published a study in 2004 investigating a user's ability to use pressure-sensitive styluses to perform discrete target acquisition tasks. Their findings show that users were not able to effectively perform pressure selection without visual feedback unless they had received at least an hour of training [3]. Their findings regarding 3D interfaces establish that assessing the performance 3D-Touch in terms of its ability to set a parameter entails providing the user with visual feedback. This led to designing an rating interface in which the mapping of pressure to a concrete value in a range was explicit. The implemented interface is shown in Figure 1 below.

The rating bar is centered within the interface, and takes up more space than any other component on the screen. A prompt that indicates which value should the user reach is placed above the rating bar, and the a pressure sensitive "Rate" button is centered horizontally within the interface and below the rating bar. This positioning allows it to be easily reachable with a thumb regardless of the user's handedness. Additionally, the current value of the rating is displayed to the right of the rating bar. The findings by Ramos et. all determined the rating bar's sizing and position with the objective of providing the user with clear feedback regarding how the amount of pressure is being mapped to the rating value.

Another notable design decision is the idea of giving different rating prompts. Early testing of this design seemed to show that achieving a low rating value is easier than reaching this figure needs to be big enough to read the text.

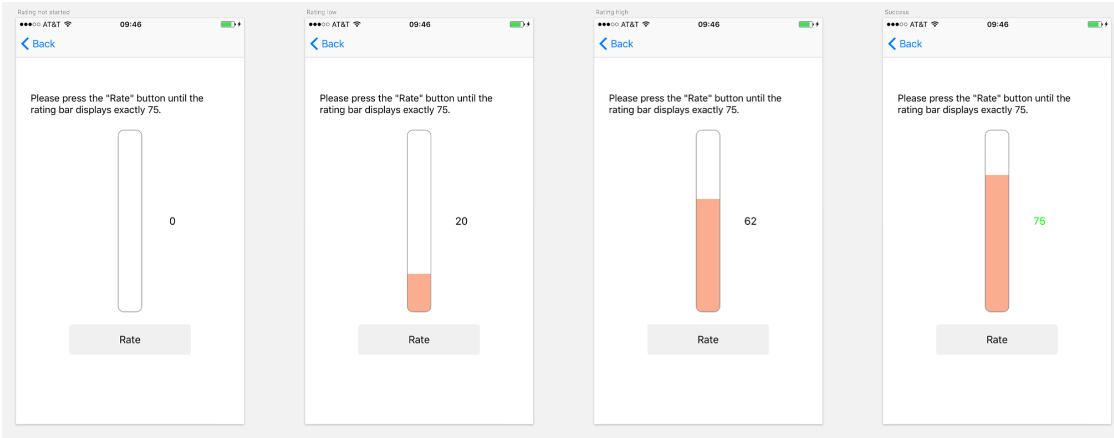

Figure 1: 3D version of accuracy experiment.

higher values. Therefore, the rating task was split into three different rating targets: (a) 25, (b) 50 and (c) 75. This design allows for determining whether a 3D-Touch interface would make achieving low rating values easier than reaching higher values. Further, most scenarios in which the user is asked to input a rating value do not have a range as large as the one in this experiment which goes from 0 to 100. Figure 2 depicts the rating interface of Apple's App Store in which there are only 5 possible rating values, and the user can either tap or slide from left to right to set their desired rating. This led to the idea of implementing prompts that urge the user to reach a value within a range of the target value. For instance, if the target value was 75, a valid rate input would be in the range from 72 to 78. Using this range idea serve the purpose of breaking down the rating into concrete pressure areas that are similar to the star rating concept, since in the latter the exact input value is mapped into either one of the 5 stars. Note that this "range" concept was made clear to the user before performing the rating task by updating the prompt value to display either:

1. Please press the "Rate" button until the rating bar displays exactly 75.
2. Please press the "Rate" button until the rating bar is between 72 and 78.

Figure 1 depicts the design of a 3D-Touch enabled rating system that addresses the user necessity for constant visual feedback, displays either an "exact value" or "range" prompts, and has a non-pressure enabled alternative (shown in Figure 3). In order to test the accuracy of 3D Touch users will be given the task of creating a new rating that satisfies the given prompt:

1. When the application starts up, the user sees a "Start" button, and he or she is told to press it whenever they are ready to begin and whenever they see the "Start" button appear on the screen again.
2. User reads the prompt to find out whether they must aim for an exact value (e.g. 50) or a range (e.g. 47 to 53).

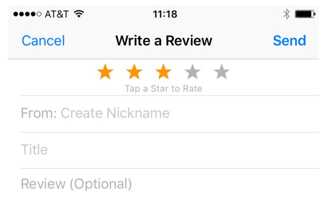

Figure 2: Apple Store App rating interface.

3. User begins to apply pressure on the start button, and the rating bar fills up with a value proportional to the amount of pressure they have applied on the screen. This step marks the beginning of a user's attempt to reach the target value.

4. When the user releases finger from the rate button, the highest value achieved is set as the rating value.

5. The system checks whether the user has meet the prompt conditions. If so, the rating value turns green to provide success feedback and the user returns to a screen with a start button, and upon pressing it they must perform the same task but with either a different target value or target type ("range" vs "exact").

6. A final screen telling the user that they are finished appears when the use has finished all the available tasks for each treatment: 3D-Touch or slider.

The non 3D-Touch version of this application (shown in Figure 3) operates in the same fashion, but the user needs to swipe from left to right on a slider that has the same dimensions and location as the "Rate" button from the 3D-Touch version of the experiment. There is an important difference that needs to be addressed regarding the implementation of the 3D-Touch and slider interface. An attempt to achieve a given rating on the 3D interface is recorded every time the user applies pressure and then releases from the "Rate" button. When the user first presses on the screen, the start time for an attempt is marked and, when the user releases his finger from the screen the end time for an attempt is determined. Therefore, the attempt time is determined by subtracting the end time from the start time and the rate value that is set corresponds to the amount of pressure the user applied. However, on the slider version of the interface an attempt's start time is marked when the user first moves the slider and the end time when the user releases his finger from the slider. If the user did not reach the target, the user must move the slider until it is on the target position and the next attempt is recorded. The difference is that in the slider case the user does not always have to start from 0, which contrasts with the 3D-Touch version of the interface. Another possible issue occurs if the user overshoots and aims to reach the target value by slowly releasing pressure from the screen. The implementation only logs the maximum value reached from when the pressure began to when the user lifted his finger from the screen. In other words, the rating value can only be reached by applying additional pressure on the screen and not by releasing pressure: if the user overshots the rating, he or she must start over from zero.

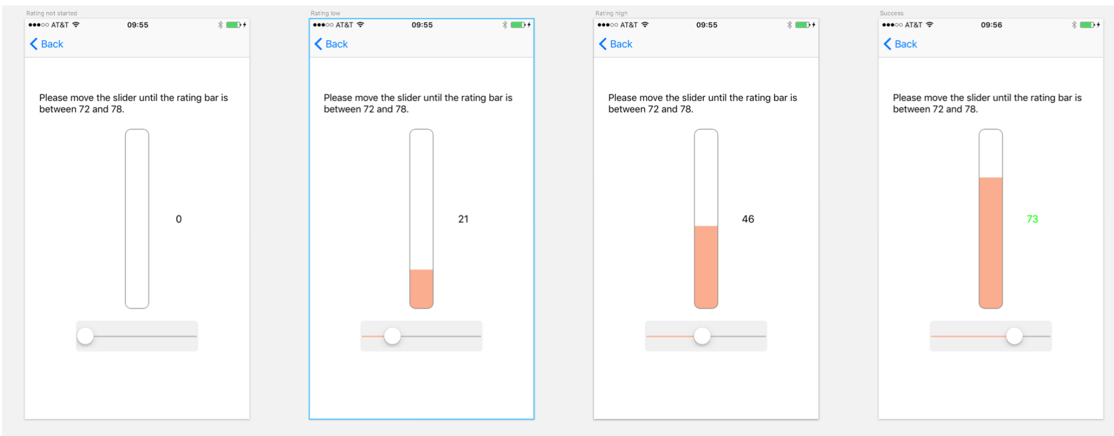

Figure 3: Swipe-based rating system.

3.2 3D-Touch Interactions and Cognitive Overhead

iOS 10's new notification systems allows developers to extend what a notification is in complex ways. For instance, a user no longer needs to open the iMessage application when he or she receives a text message. With this new system, a user can simply apply pressure over the notification until a complete view of the notification along with a keyboard appears. Note that this interaction is not based on a screen tap, but on the user applying pressure over the notification pop-up until the device's haptic feedback mechanism confirms that the current view has been locked (see Figure 4). 3D Touch adds an invisible layer to the interface that can be accessed by applying pressure on the screen. This feature, in combination with the new notification environment provides a more streamlined access to the content (not the features or tasks) of an application. Further, the ability to 3D Touch a notification in order to reach a pop-up allows the user to perform a task on a different application than the one he or she is currently using without having to leave the application's notification environment. In a sense, 3D Touch provides users with the ability to easily go back to what they were doing without breaking the flow of their previous task.

Additionally, 3D touch can be compared to the "control" button on a Mac computer. 3D Touch lets you peek a little deeper inside applications and find out something interesting–rather than just clicking on them to get a menu. This idea is important given the that the nature of mobile applications is more about consuming content than performing tasks. This idea of "consuming content" is fitting for 3D touch—it allows the user to take care of

Figure 4: iMessage notification. Left: before pressure is applied. Right: after pressure has been applied, and the notification has been locked on the screen.

whatever content needs their immediate attention without leaving their current screen. It is important to emphasize that consuming content refers to the ability to scroll through all the messages from a previous conversation while still on the 3D Touch enabled notification environment, and thus being able to return to the previous application without having to navigate.

The objective of this section is to test whether 3D Touch allows for a less disruptive notification system. Cognitive overhead is defined as how many logical connections or jumps you brain needs to understand or contextualize the present-on-screen software [1]. In other words, it addresses how much time it takes for a user to refocus on their previous task after they have interacted with a notification. This experimental section will require the user to be engaged in a not-so-trivial task when a notification will overlay over the current screen. To this end, a simple game was designed to keep the user engaged before a 3D-Touch enabled or a "traditional" 2D notification appears on the screen. This game has the objective of allowing us to test the effects of 3D-touch enabled notifications on cognitive overhead. It's goal is to select shapes according to the following rules:

- Green + Square
- Yellow + Triangle
- Red + Circle
- Blue + Diamond

The game is simple enough so that it does not have a steep learning curve. Additionally, the rules of the game are always available for the user to see. When the game loads, the

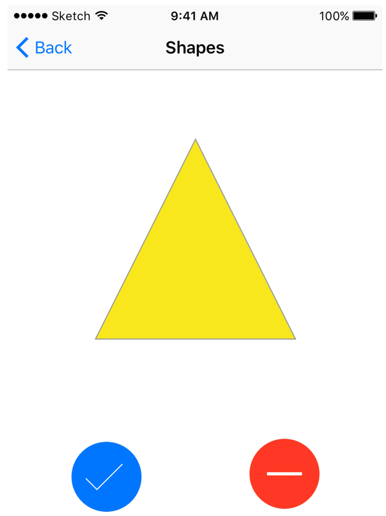

Figure 5: Main screen of the “Shape Game”. The blue button confirms that the shape matches the rules above, and the red button does the opposite. This triangle is a “correct” shape.

user sees a screen containing a single shape (shown in Figure 5). The shape on display may or may not fit the criteria above, and the goal is for the user is to tap on the “Confirm” or “Reject” buttons displayed below the current shape to determine whether its a valid shape according to the rules. This game can be divided in three stages:

1. A shape appears on the screen of the device.

2. The user mush press the “Reject” or “Confirm” button to assert whether is a valid shape or not.

3. User repeats steps (1) and (2) above to play the game.

After the user has had some experience playing the game, a notification will appear on the screen prompting the user to either tap or press on it. This experience is necessary so each user has at least some familiarity with the rules, and he or she is not simply guessing. The instruction that prompts the user to deal with the notification produces two different scenarios:

- (3D-Touch Notification) In this case, the user interacts with the notification by applying pressure on it. Then, the user is instructed to perform a task within the notification. An example task is to rate a meal that was just published on a social meal tracking application. After the meal is rated, the user taps outside the notification to return to the game. The 3D version of this application is shown in Figure 6.

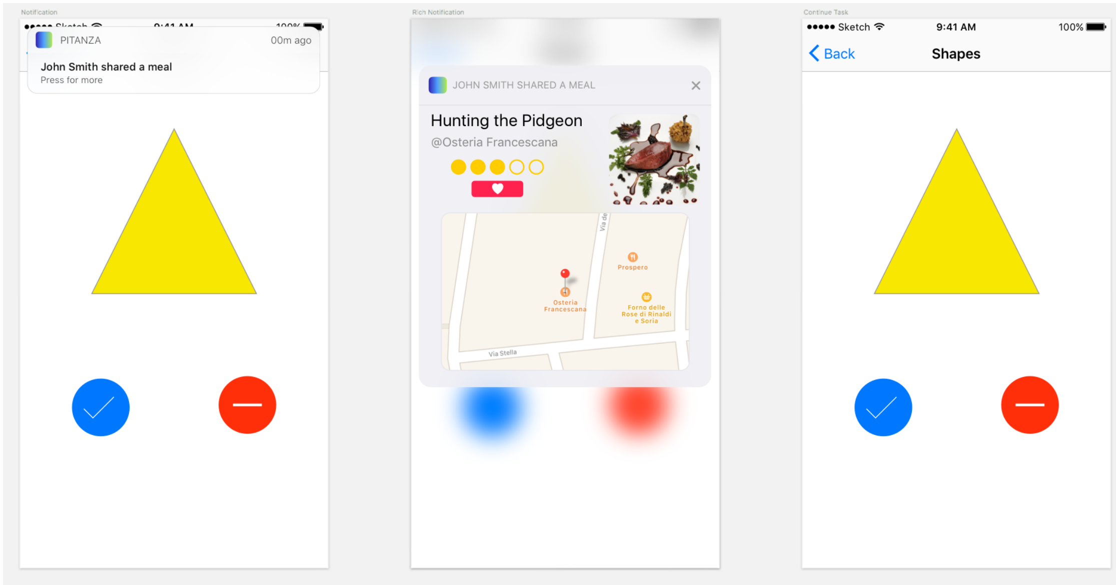

Figure 6: Each user goes trough the following screens during the "cognitive overhead" experiment. Note that the notification overlays the game screen and leaves it as a blurred background. When the user interacts with the notification, he or she is returned to the leftmost screen.

- (Traditional) In this case, the user taps on the notification to access the "Rate a Meal view" within the application that sent the notification. At this point, the user is instructed to rate the meal. In order to return to the game, the user must tap on the top left corner of the screen. This navigation concept is embedded within iOS, and it's implemented in virtually all applications–it will not be surprising for a user familiar with the iOS ecosystem. This interaction is shown in Figure 7.

After the user has returned to the game screen, we will measure how much time the user takes between interacting with the notification and confirming or rejecting the shape currently present on the screen. This assessment will be done for both scenarios: (a) when the user interacts with the notification overlay (3D-Touch), and (b) when the user has to navigate back from the meal tracking application in order to continue playing the game.

The task above (shown in Figure 6) requires two features of 3D Touch that may be unfamiliar to a user: (a) applying a gradual amount of pressure until the pop-up locks and (b) navigating back to the previous screen by tapping on the blurred side areas. In order to mitigate this issue, before start of each experiment the user will be shown the following video. This short video provides an overview of the main features of 3D Touch and how they apply to the iOS operating system. Given this training, it is possible to measure cognitive overhead by determining the time between closing the notification and how long it takes for the user to make the next move in the "Shape Game". Part of the time difference may arise from the easier navigation associated with this 3D touch interaction; however, a one tap "back" button is located on the top left of the screen for all applications. It's important to emphasize that one tap is also the command required for exiting the 3D-Touch expanded view of a notification. Figure 7 shows the interaction path for the user of a 2D version of

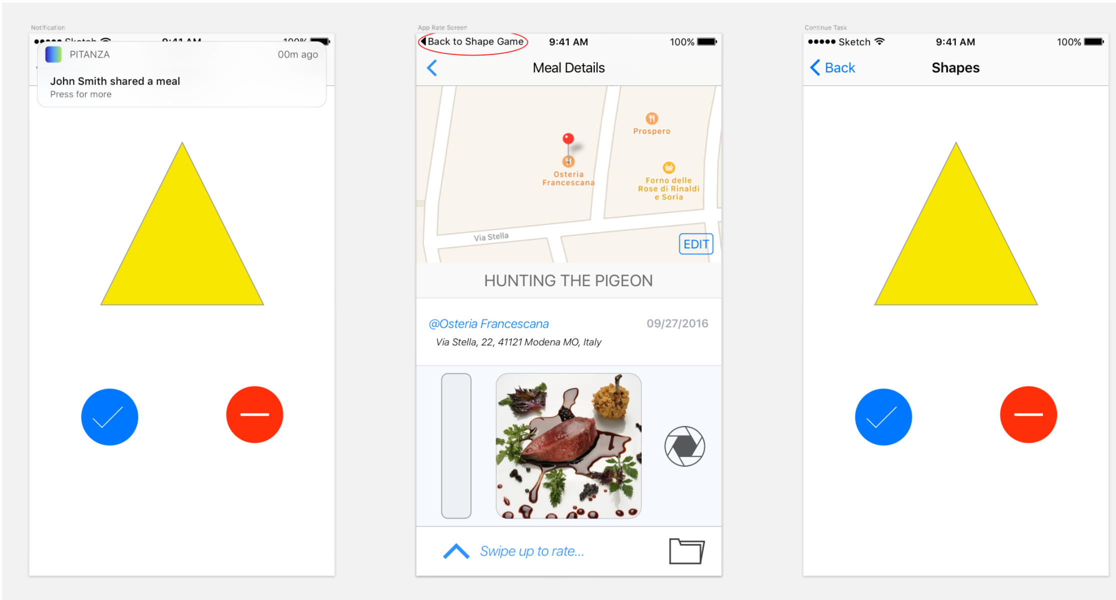

Figure 7: Shape game and notification in "Traditional" 2D notification interface. Note that the button that allows for navigating back to the previous screen has been circled in red for clarity.

this interface. In this version, the user must tap on the "Return to Shape Game" button to continue playing the game. Therefore, since it is almost as easy to navigate back in both interfaces, the time difference, if any, will serve as a measurement of cognitive overhead.

3.3 Performance of 3D-Touch as an Error Recovery Mechanism

3D touch interactions tend to be marked by the usage of a pop-up that gradually grows as a user applies more and more pressure on the screen. This provides a "preview" environment in which the user can abort the 3D action without being forced to navigate back to the previous screen. This idea will be tested by implementing the interface shown in Figure 8, which contains three 3D touch enabled buttons. As the user starts to apply pressure, a pop-up of a given color begins to grow. The user is given the task of finding a specific color, and we will measure the time whenever they choose the wrong button to see how fast it takes for them to recover and pick the right one. The task for this experiment is the following:

1. User beings to apply pressure on one of the buttons on the bottom part of the screen–labeled A, B and C. The user has been instructed to find a particular color.

2. As the user applies pressure, a pop-up of one of the predetermined colors appears.

3. If the user "locks" the pop-up view, a button appears that can be used to confirm the choice.

4. If the user taps outside the colored box after locking the view, he or she is returned to the initial state in which none of the boxes have been opened.

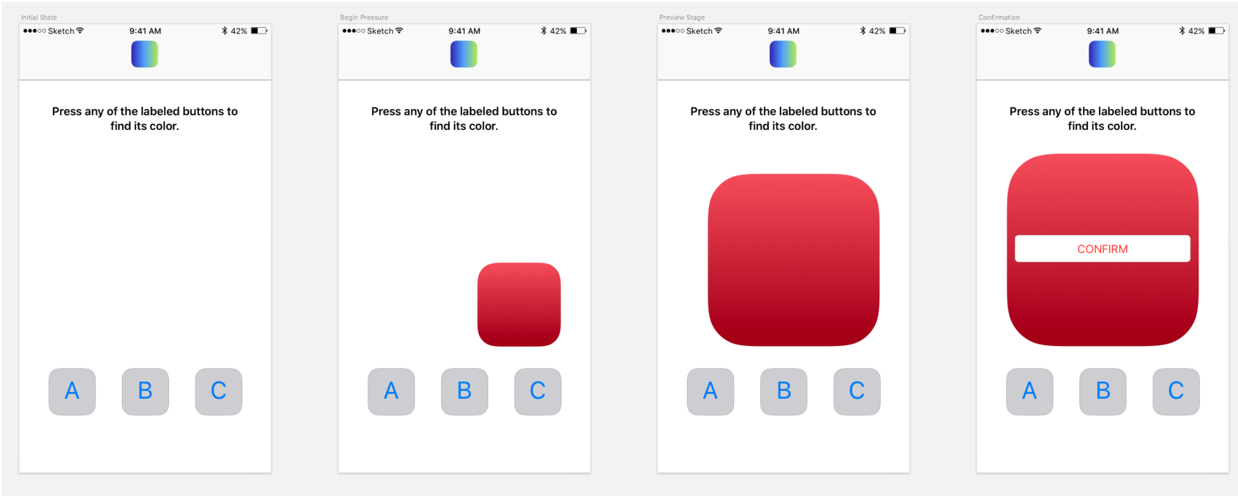

Figure 8: 3D-Touch Error Recovery Experiment

The non-3D version of this interface will be implemented such that a tap on one of the buttons opens the rightmost screen on Figure 9. From this view, the user can simply tap to confirm the color selection—appropriate feedback will be given to the user. The metric for this experiment will be the time that it takes the user to recover from choosing the wrong button. The maximum amount of different buttons that the user can try is two, and thus each test subject will be classified with his or her number of tries. This allows us to determine the error recovery time regardless of whether the second or third choices are the solution. The goal of this experiment is to test the "preview" environment of 3D Touch interactions as an effective error recovery mechanism; for instance, by preventing unwanted navigation.

4 Experiment Results

4.1 Accuracy

The 3D-Touch accuracy experiment was administered to 11 Union College students. Table 1 above summarizes the experiment results. For all target values (25, 50 and 75) the number of attempts was around 9 when the user was given the task of reaching an exact value. The number of attempts stayed at around 2 for all cases in the slider treatment. Note that in the 3D-Touch implementation, the number of attempts increases slightly when the user is asked to reach a number between 72 and 75–it increases from around 3 to over 5.5 attempts. This may suggest that numbers higher in the range are harder to achieve. The total figures represent the average number of attempts and average time taken to perform each task for both treatments when either aiming for an exact value or a range. The number of attempts in the 3D-Touch "exact value" treatment is the highest with 9 attempts. With the same treatment but aiming for a "range", the number of attempts went down to approximately 4. This numbers are substantially higher than their counterpart for the slider treatment. Users were able to set the slider to an exact value in about 2 attempts and in approximately 1 attempt if they were aiming for a "range" value. Further, the average time taken to perform

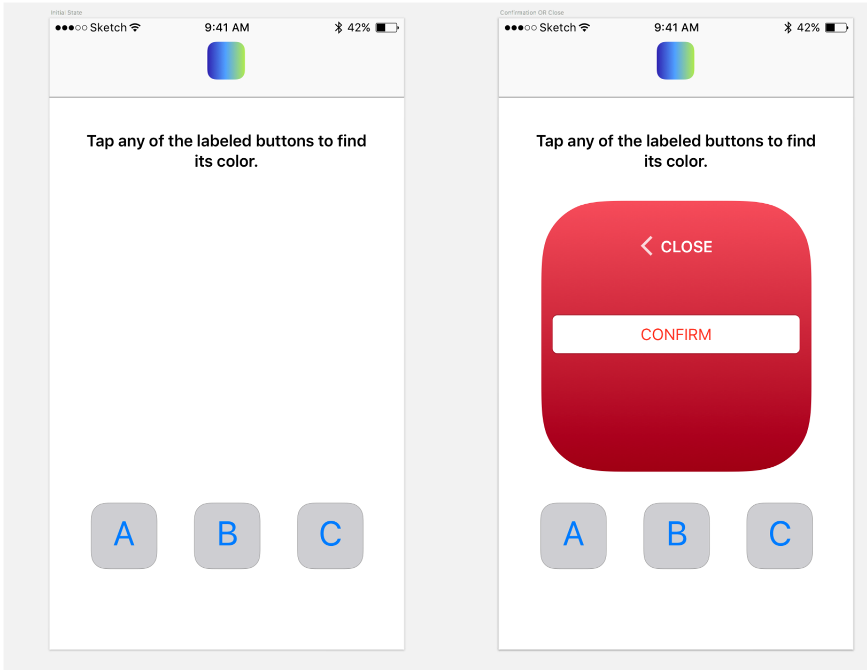

Figure 9: Traditional "2D" Error Recovery Experiment

any of the tasks was the highest on the 3D-Touch "exact value" treatment. The users took an average of 19 seconds to perform the task. It is important to note that the average time taken to perform the "range" task is very close for both treatment: users took about 4 seconds to perform the slider task and approximately 5 seconds to perform the 3D-Touch version of the task.

In addition to collecting data regarding the number of attempts and the time taking for each attempt, each participant was asked to fill out a small survey with basic demographic information as well as familiarity with 3D-Touch. The survey asked the following questions:

Summarize the results of these! Don't just give the reader the questions.

1. Are you a smart-phone user?

2. If yes, what is your mobile operating system of choice?

3. Do you keep up to date with Apple products? This question prompt the user to enter a value between 1 and 5.

4. The participants were asked to rate their familiarity with 3D-Touch on a scale of 1 to 10.

5. Participant's age and gender was also collected.

The objective behind administering this survey was to address whether previous familiarity with 3D-Touch improved the performance during the accuracy experiment. Table 2 shows how each participant performed and as well as their reported degree of familiarity with 3D-Touch. Most users ranked their familiarity with 3D-Touch at 5, which the midpoint for the given range from 1 to 10. There does not seem to be a relation between the reported

	Slider		3D-Touch	
	Target	*Range*	*Target*	*Range*
25	2.000 0:00:06.888	1.182 0:00:04.275	9.182 0:00:17.916	3.364 0:00:02.367
50	1.727 0:00:05.944	1.091 0:00:03.958	9.455 0:00:19.690	3.100 0:00:09.117
75	2.182 0:00:09.061	1.000 0:00:04.188	8.545 0:00:19.470	5.545 0:00:04.837
Total	1.970 0:00:07.298	1.091 0:00:04.140	9.061 0:00:19.025	4.003 0:00:05.440

Table 1: Average time taken (ms) and number of attempts per target (range or exact) value for both the slider and 3D-Touch treatments.

experience with 3D-Touch and their performance on this experiment. For instance, user number 7 reported a familiarity of 7, and yet he or she took the longest time to complete the exact value task. Similarly, user number 3 reported a familiarity of 7, and his or her time is second to longest. In addition, user number 4 reported a familiarity of 1 and his or her time for performing both tasks is on the lower end of the spectrum. Note that the number of attempts for this particular user is approximately 15 and 11 for both tasks. Although the time taken is low, the number of attempts is on the higher end of the range. This seems to suggest that this unfamiliarity with 3D-Touch led the user to try tapping on the screen instead of pressing. Further, given the reduced number of users and the fact that most of them were iOS users, it is complicated to extrapolate the effect of familiarity with 3D-Touch in user performance on this experiment.

Participant	Mobile OS	3D-Touch Familiarity	Exact Value Attemps	Exact Value Time	Range Attempts	Range Time
1	iOS	4	5.667	0:00:17.638	1.000	0:00:00.966
2	iOS	3	4.333	0:00:29.494	1.000	0:00:01.609
3	iOS	7	14.667	0:00:30.973	2.667	0:00:02.578
4	iOS	1	14.667	0:00:12.582	11.000	0:00:08.156
5	iOS	5	13.333	0:00:04.874	13.333	0:00:03.947
6	iOS	5	15.333	0:00:10.298	4.667	0:00:21.686
7	Android	7	10.667	0:00:37.359	3.000	0:00:04.011
8	iOS	5	3.333	0:00:16.764	1.000	0:00:01.270
9	iOS	4	8.333	0:00:20.329	2.000	0:00:05.218
10	Android	5	7.000	0:00:24.583	2.000	0:00:06.123
11	iOS	5	2.333	0:00:04.385	1.667	0:00:01.674

Table 2: Average number of attempts and average time taken for all target values (3D-Touch) mapped to survey results per participant.

You don't need to give me per-subject data. Summarize. How many iOS vs. Android? Etc. And you are missing any statistical analysis on your times to see if there is any statistical significance.

5 Conclusions and Future Work

Assessing the impact of 3D Touch on user experience is a complicated task. This new technology allows developers to craft interfaces that are pressure-sensitive; however, this sensitivity does not necessary grant a better user experience. The results on Table 1 suggest that 3D-Touch can be an alternative to a traditional slider interface for setting a rating of the type shown in Figure 2. As stated above, the range treatment for both interfaces aimed to simulate this kind of rating system. The results show that users were able to perform the task in about 5 seconds for each case–a little less for the slider interface. However, the number of attempts in the 3D version is substantially higher. This findings suggest that 3D-Touch for this kind of task would have a negative effect in user experience: the users are able to perform the same task in the same time but in a more "trial-and-error" manner. Although the 3D-Touch interface was more challenging to operate, the users appeared substantially more engaged on the task and a few of them suggested the addition of 3D-Touch to an iOS video-game. The results on Table 2 entail that there is barely any learning curve to 3D-Touch. As aforementioned, user 4 which was barely familiar with the interface was able to perform the tasks rather quickly by performing a high number of attempts until he or she had a proper understanding of how the interface worked. Unfortunately, I have not been able to implement and test the experiments that aim to assess cognitive overhead and the usage of 3D-Touch as an error recovery mechanism. These two experiments remain as future work. Yet this findings in terms of the usability of 3D-Touch as well as the user engagement that I observed while users were interacting with pressure sensitive interfaces suggest that users enjoy 3D-Touch interfaces. 3D-Touch may not be suitable for an interface that requires performing a very precise task, but it may work very well as part of a video-game interface. An experiment that assesses the user-experience of 3D-Touch enabled video-game environment will complement the experiments mentioned above in order to address the impact of 3D-Touch in the iOS ecosystem.

References

[1] Nikunj P Dalal, Zane Quible, and Katherine Wyatt. Cognitive design of home pages: an experimental study of comprehension on the world wide web. *Information Processing & Management*, 36(4):607–621, 2000.

[2] Yeon Hwa Kwak, Wonhyo Kim, Sungkyu Seo, and Kunnyun Kim. Comparison of two types of tactile sensing layer in touch screen panel for force sensitive detection. In *SENSORS, 2015 IEEE*, pages 1–2. IEEE, 2015.

[3] Gonzalo Ramos, Matthew Boulos, and Ravin Balakrishnan. Pressure widgets. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 487–494. ACM, 2004.

You don't mention about how these 3D-touch aspects are not exclusively tied to 3D touch interfaces. One could implement error-recovery and an "overlay" mode in 2D just as easily. It's just that these happen to be part of the 3D touch experience. It's important to acknowledge that you are not testing something that is 3D touch exclusive.