A Comparison Between a Virtual Joystick and Fling Gestures as Control Inputs for Mobile Gaming

Zian Liu
Dept. of Electrical Engineering and Computer Science
York University
Toronto, Ontario, Canada M3J 1P3
vice9608@my.yorku.ca

ABSTRACT
This study seeks to determine which control method is more suitable as the default built-in input for touch screen device gaming. Eight participants were recruited to compare the performance of the virtual joystick and fling gesture on touch screen Android devices. A simple brick shooter game with two different brick motion speeds was used to examine each test condition's completion time and accuracy. The virtual joystick method had the same average accuracy with fling input but was 15% faster than the fling input in terms of the meantime. The slow movement difficulty level took 31% less time to complete than the fast movement difficulty level; the slow movement difficulty level also proved to have higher accuracy. Joystick at the easy difficulty level had the shortest completion time and highest accuracy among all test conditions whereas the same input at hard difficulty level had the longest completion time and lowest accuracy. Most participants preferred the virtual joystick and noted that the fling gesture is difficult to maneuver.

Keywords
Virtual joystick, fling gesture, direction control, mobile games

INTRODUCTION
With the widespread adoption of smartphones, gesture control has become common for mobile users. In gesture control, gestures perform different functions. Some of them are simple, like a double tap on the screen, while others are fancier, like using a custom gesture to unlock an *iPhone*. Nowadays, more advanced gesture controls are developed for virtual reality and other high-tech devices.

For most gestures, developers only need to combine a function with a suitable gesture so that it makes sense to users who can easily understand what will happen when they use that specific gesture. However, what about efficiency? How precisely can a gesture be performed? It may be unnecessary to test this for daily tasks, but in the area of gaming, efficiency of gesture is important.

Although the video game market has matured and there are plenty of design principles or frames to follow, mistakes are still occurring in the field. For example, in 2017 Apple keynote event, a game named "*Sky*" was first revealed with the new *iPhone 8*. It was made by a famous console games studio with support and improvement from Apple. Everyone was looking forward to its release, but after the beta version launching, most game media described the game experience was painful [2]. As a third personal role-playing game, the developers abandoned traditional virtual joystick control like other games, creating a brand-new gesture control system that allowed players to use both single-touch and multi-touch. However, most gesture combinations they created left players uncomfortable and confused even after a few hours adjusting to the controls. For example, in order to make their character move forward a step, the player had to flick on the screen. Imagine a virtual world where users need to keep flicking again and again to explore: the larger the world map, the worse game experience on the player. Since almost every single user testing the beta was complaining about this interaction, and the self-made gesture control system was the one of most important areas of content the game's makers wanted to include, the game still hasn't been released after two years, with no updates.

Do mobile game companies do scientific research before implementing a new gesture control system? Perhaps not, because there is one easy and cheap way to test a game: let players try a beta version and just drop the most unpopular designs. The case of *Brawl Stars*, a multiplayer mobile competitive game, demonstrates this tactic. In its beta period, there were two control methods that allowed users to attack. The first method was tap and swipe (or drag to shoot); players click the screen to control their character's movement and direction, then tapped the screen and swiped to shoot a bullet to character's forward direction. The second method that optimized up to now is using double virtual joysticks as shown in Figure 1. A "virtual joystick" is a common mobile game input that controls the player character or an object on the screen. It can detect where a button is dragged to, and what strength is applied to the button. In *Brawl Stars'* double joysticks mode, the left joy stick was for movement and the right one was for aiming and shooting. After almost two years of beta testing before the game was fully released at the end of 2018, the tap and swipe mode was removed from settings, and only the double joysticks mode was left [6]. The game hasn't revealed any details or data about why they removed one input mode, even though some people are still requesting they add the tap and swipe mode back.

Figure 1. *Brawl Stars* double joysticks mode.

To figure out the performance difference between two *Brawl Stars* modes, this paper was designed. Future mobile games can use the result to improve their interaction system.

Related Work

Bragdon et al. [4] tested the speed and accuracy of three touch-screen control techniques: bezel gestures, hard button-initiated gestures, also soft buttons. Rectilinear and free-form path-based gestures were implemented to compare performance. Participants were asked to memorize the gestures in the question and then complete the task requirement with those different techniques. Various levels of motor activity and distractions were also added to simulate more scenarios. The result showed bezel gestures always had the best performance in either accuracy test or speed test and the hard button had the worst performance. As shown in Figure 2, for overall performance bezel marks' completion time was 12.17% shorter than soft button. However, there was no significant difference about overall accuracy between hard button marks, bezel marks and soft buttons which were 93.5%, 92.5% and 91.2% respectively. Furthermore, bezel gestures also had better performance in the walking environment and overperformed soft button in different distraction cases which represents it could be articulated eyes-free.

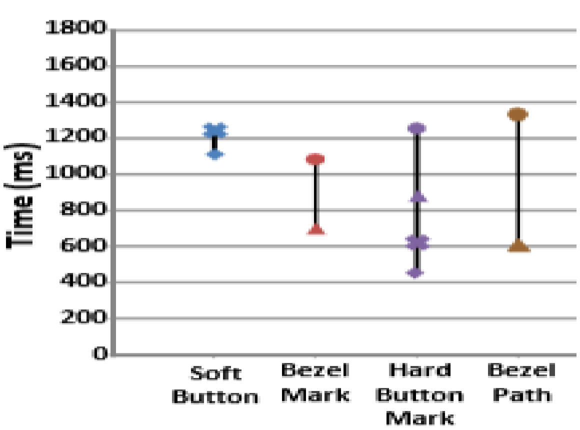

Figure 2. Task completion time breakdown by input [4].

Abubakar et al. [1] designed a 3D game for Android named *Curious Jojo*. It was used to spread the go-green concept to the general public. Players collected recycled and reusable items in a 3D game world while negotiating a few obstacles. Figure 3 shows the two virtual joysticks used in the game. They are virtual 2D buttons, like those in *Brawl Stars*. The left joystick is for movement and the right joystick maybe used for interacting with game-world objects. The latter was not discussed in this paper. The game was assessed by the public at an exhibition. No negative feedback was received about the game's functionality.

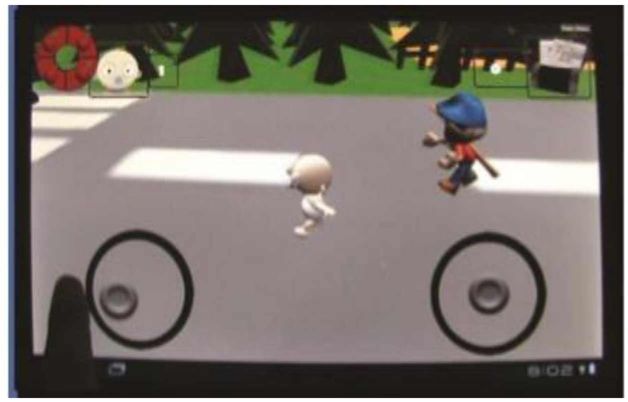

Figure 3. *Curious Jojo*'s GUI [1].

Baldauf et al. [3] conducted an experiment to compare the performance of directional buttons, a directional pad (which uses an embedded soft d-pad), a virtual joystick and tilt control on a smartphone. Participants first did a formal control test using these techniques to move a character from the center of the screen to the center of a target, which had a random size, distance and orientation. They also tested the performance of the techniques in two commercial games: *Pac-Man* and *Super Mario Bros*. In the formal control test, the directional pad had the shortest completion time, since participants needed only to memorize the approximate position of this target with this technique, whilst they had to confirm their input for the other techniques. For the two games, the directional pad got the highest average game score. In these two games, players can move in only four directions (up, down, left and right). It is possible that the joystick would have had the best score in a more complex game. In addition, because the tilt controller is difficult to use, it always had the worst performance.

Li [7] built an open-source gesture toolkit, which programmers can use to design their own custom gestures for either a phone system or an application. It had two main components. The first are gesture overlays, which collect each gesture event within a relative bounding box and path distance and then dispatches it to the software. The other important component is gesture recognition. This component is used by the toolkit to learn how each user writes the English alphabet. These gestures for letters are stored in a library. With the toolkit, programmers can upgrade applications that use gestures to make them capable of doing more complex tasks.

Jennings et al. [5] examined a few input methods for touch screens, looking at error rates and the time to complete a task. Participants were requested to use arrow buttons, a flick gesture, a navigation bar or a scroll bar on either a phone or a tablet. They were asked to scroll the screen to find a target. The participants always took less time on a phone than on a tablet, regardless of the input method. For the flick gesture, the difference in time between the two device types was less than 3%. The navigation bar took the least amount of time for both screen sizes, since it is well suited for this kind of task. The flick gesture had the longest mean time on both devices, taking 101% longer than a navigation bar on the phone and 81% longer than a navigation bar on the tablet. The flick gesture also had the highest mean error rate on the phone and the second highest on the tablet, which indicates that it is not an efficient way to scroll for something on a touch screen. However, the flick gesture required the smallest amount of effort for scrolling compared to the other input methods. On average, it required only 28% more effort than the optimal method.

Zaman et al. [8] compared touch-screen controls and physical controls for game input. The performance of players in a video game when using a virtual directional pad and other soft buttons on an *iPhone 3G* was compared to their performance in the same game on a *Nintendo DS*, which has physical buttons. The average number of deaths during the game when using touch-screen controls was 150% higher than when using physical controls. This means that the players made many more errors, like missing a jump, with the virtual buttons. The completion time of a game task was 25.4% shorter on a *Nintendo DS* compared to the *iPhone*. This study illustrates that since a self-made virtual joystick and buttons may perform worse than expected, we should focus on testing and optimizing the efficiency of controls when building an application.

METHOD

Participants

Eight unpaid participants were recruited for the experiment. All participants are university students around 18–23 years old. Two of participants are female; six are male. Two participants are left-handed. Each participant has few hours in using phones every day.

Apparatus

The test was conducted using a simple self-made Android application that simulates a brick shooter game as shown in Figure 4. There is a rectangle on the top of the screen moving continually back and forth from left to right. A preset ball is placed at the bottom of the screen that so players can shoot at the moving rectangle. In the game's settings, the rectangle's speed can be set at either slow motion, which is easier to predict and hit, or quick motion, which is harder. Two input modes were designed to examine accuracy and speed for the task, flick mode and virtual joystick mode. The black circle in Figure 4 was the soft button for virtual joystick. Both modes apply the same speed to the ball when the right gesture is detected, and each trial was tested to record the completion time when ball hits the brick five times. After the gesture occurs and the ball hits the screen's edge or the rectangle, the ball will be respawned to its original position.

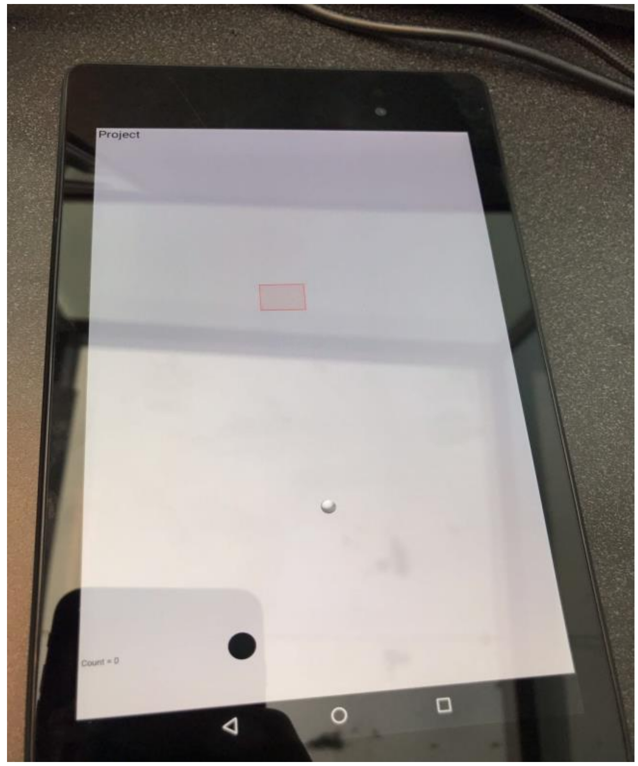

Figure 4. Experiment application interface.

In flick mode, users first need to tap the screen close to the ball's spawn position, and then swipe and release in the direction they want to aim the ball at the moving rectangle. The application records the swipe gesture's direction and instantly give the ball a speed in that specific direction. If the ball hits the rectangle, players score a point. After the score is equal to five, a result page pops up that includes completion time, the number of miss and accuracy (the percentage of attempts resulting in a successful hit).

Joystick mode is similar to flick mode, but instead of tapping and swiping, users must use a virtual joy stick on the screen to control the ball's direction. The application also records the direction that the virtual button on the joy stick is dragged. When the virtual button reaches a specific range from the joy stick's center, the application will automatically shoot the ball. The same results page will appear after the game is finished.

Procedure

After the participants arrive, they were told about how the application works and how to complete the experiment task. Everyone was given the same amount of time to practice, allowing them to get familiar with the ball's shooting speed in both modes and learn to hit the rectangle later in the trials. Participants were divided into four groups for counterbalancing, each group did the experiment in a different order for control mode and rectangle speed. In each trial, participants were asked to sit and do their best to finish the task like in Figure 5. They must care about not only their accuracy, but also the number of attempts they are able to generate. Each of them spent about 10 minutes to finish the whole task. At the end of each trial, the result page was collected for later analyses.

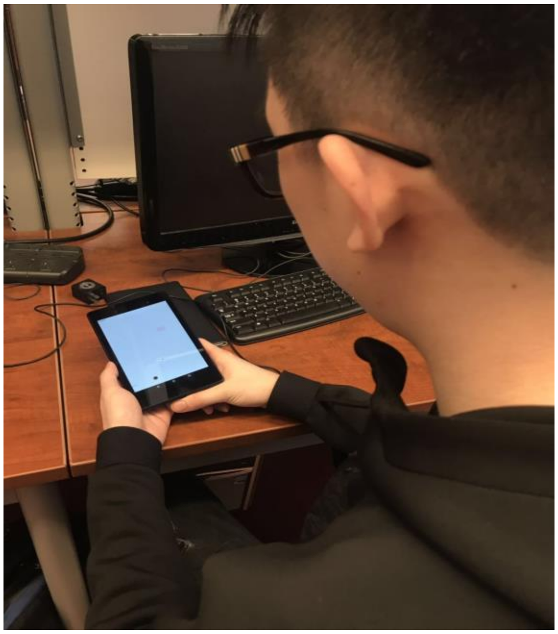

Figure 5. A participant was doing the task.

Design

The user study employed a 2×2 within-subjects design. The independent variables and levels were as follow:

- Game control mode (flick, virtual joystick)
- Game difficulty (quick target (hard), slow target (easy))

The dependent variables were completion time, hit percentage and the number of miss.

In order to offset learning effects, participants were divided into four groups and then each group followed the order in a 4×4 balanced Latin square so that no group was tested in the same sequence for independent variables.

Participants did every game mode with each difficulty 5 times, so there were 8 participants × 2 modes × 2 difficulties × 5 trials = 160 trials.

RESULTS AND DISCUSSION

The grand mean for completion time was 8.96 s. The shortest time recorded in joystick mode was 2.73 s with a 100% accuracy and the longest time was 33.93 s with a 29.41% accuracy.

Figure 6 presents the time results for two control methods at both difficulty levels. By input method, the mean completion time was 8.32 s for joystick and 9.6 s for fling input. The fling method was about 13% slower than the joystick input. By difficulty levels, the result was 10.62 s for the hard difficulty level and 7.29 s for the easy difficulty level, which was 31% faster than the former.

The long completion time of the fling method can be attributed to the inefficient detection of gesture input when participants interacting with the screen. We observed that participants sometimes put the whole tip of their index finger on the screen for the fling gesture, which would cause the software to generate a slightly different direction for the ball than intended. This seldom occurred in the joystick mode, since participants could see the direction of the virtual button's movement from their screens.

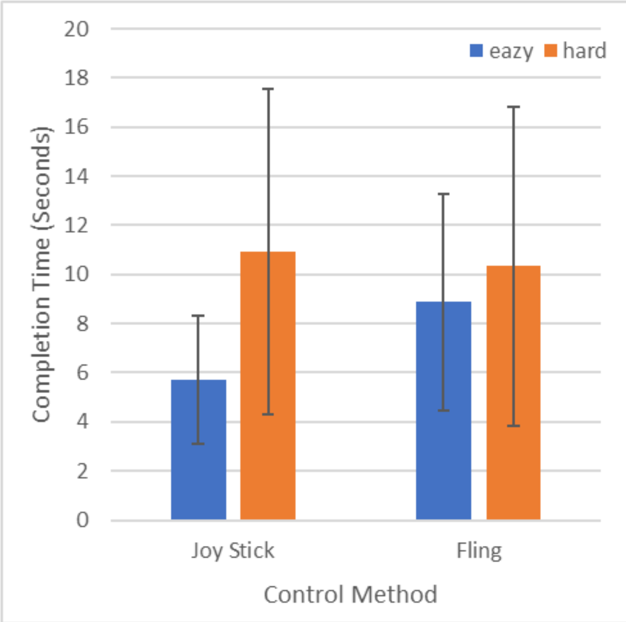

Figure 6. Time by both methods with different difficulty.

By Method × Difficulty, the mean for joystick at the easy difficulty level was 5.71 s which is the fastest time among all test conditions. It was about 48% faster than the joystick at the hard difficulty level at 10.93 s, 36% faster than the fling method at the easy difficulty level at 8.87 s, and 45% faster than the fling method at the hard difficulty level at 10.33 s.

The chart in Figure 7 indicates the completion time by trial number for each software input. There was an obvious decrease in completion time for fling gesture from 10.72 s for the first trial to 9.46 s for the final trial, which was approximately a 12% decrease. As the chart demonstrates, improvement with practice was more prominent for the fling method than for the joystick method. This is because most participants reported being uncomfortable with the fling gesture method at the beginning and required time getting used to the method.

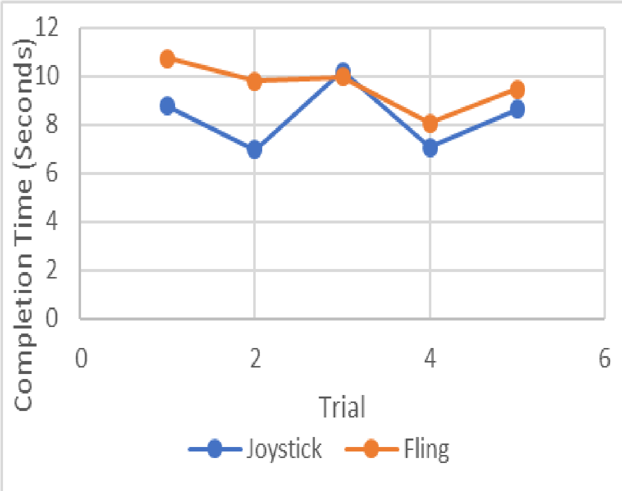

Figure 7. Task completion time by trial.

The accuracy data was calculated by the number of successful hits, which was 5 divided by the number of attempts in each trial.

By control method there was no distinct difference, the accuracy rates were 57.22% for joystick input and 56.96% for fling input. By difficulty, the mean was 62.22% for the easy difficulty level, which was higher than the hard difficulty mean of 51.96%. As seen in Figure 8, by Method × Difficulty, the joystick easy mode had the highest accuracy at 67.44%, which was 43% higher than the lowest mean accuracy of 47% recorded in the joystick hard mode. Surprisingly, the difference was quite small among test conditions with the fling method, the average accuracy didn't drop with hard difficulty. The means were 57% for the easy mode and 56.91% for the hard mode.

The accuracy result did not conclude which method was better. This is probably because when participants miss a few attempts, they begin to focus on the accuracy performance and tend to focus less on their completion time.

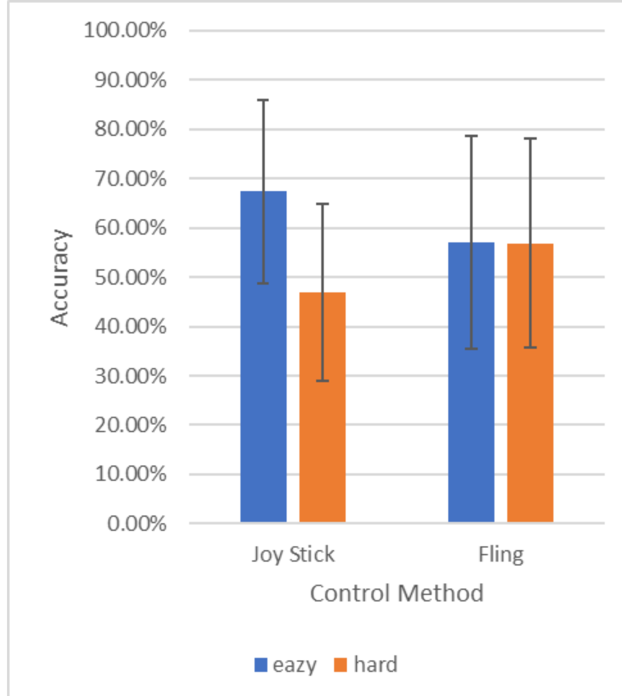

Figure 8. Accuracy performance by both methods with different difficulty.

We also recorded the learning progress for accuracy. Figure 9 shows the data of two methods over 5 trials. The accuracy of both methods has increased a little between the first trial and the last trial. The joystick method had a better improvement rate of 14%. Interestingly, the best accuracy performance for both inputs occurred during the fourth trial rather than in the fifth trial.

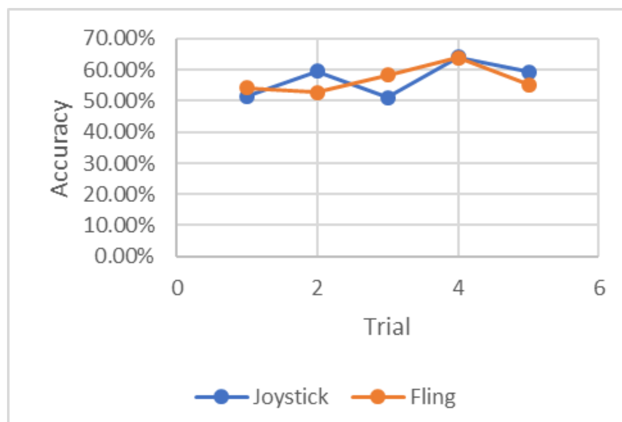

Figure 9. Accuracy by trial.

An open-ended questionnaire was given to participants at the end of the experiment. 5 participants preferred the joystick input, 2 participants preferred the fling input, and 1 participant liked both. However, both inputs received negative feedbacks, especially for the fling method. A few participants commented that the slight latency and incorrect direction occurred during the fling gesture mode testing was not very user-friendly. One left-handed participant remarked: "Since the application puts the virtual joystick to the left of the screen rather than in the middle, right-and left-handed people would have different experiences and performances when they push the virtual button to the left or the right side." Every participant reported that they felt challenged when they first entered the hard mode but got used to it by the end of the test.

CONCLUSION

In this user study, we designed a touch screen Android brick shooter game to evaluate and compare the efficiency and performance of two mobile control inputs, the virtual joystick and the fling. Each input was tested with two difficulty levels based on the speed of the brick movement.

The joystick method had the fastest completion time with 8.32 s, and the fling method was 13% slower at 9.6 s. By difficulty levels, the easy difficulty level had the fastest completion time at 7.29 s, 31% faster than that of hard difficulty level, at 10.62 s.

For accuracy, the results for the two input methods were close. The joystick method had an accuracy rate of 57.22% and the fling method 56.95%. Since it was easy to predict the collision at the easy difficulty level, the mean accuracy rate was 62.22%, which was much higher than the 51.96% recorded at hard difficulty. The joystick method at the easy difficulty level had the highest accuracy rate at 67.44% and the shortest completion time at 5.71 s among all test conditions. In contrast, the joystick method at the hard difficulty level had the lowest accuracy at 47% and the longest completion time at 10.93s.

Overall, the joystick input method performed better than the fling input method in our brick shooter application. We concluded that future work in the field should focus on modifying and enhancing the performance of fling detection. A better comparison can be made if the application's fling input works smoothly like current commercial mobile games.

ACKNOWLEDGMENT

REFERENCES

1. Abubakar, J. A., Mutalib, A. A., & Permadi, D., Design and Development of Curious Jojo©: A Go-Green 3D Game on Android. *TELKOMNIKA Indonesian Journal of Electrical Engineering, 11*(6), 2013, 3123-3129.
2. Adams, S., 'Sky: Light Awaits': Early Access Impressions, https://www.joyandsticks.com/single-post/Sky-Light-Awaits-Early-Access-Impressions, (accessed 24 June 2018).
3. Baldauf, M., Fröhlich, P., Adegeye, F., & Suette, S., Investigating On-Screen Gamepad Designs for Smartphone-Controlled Video Games. *ACM Transactions on Multimedia Computing, Communications, and Applications, 12,* (New York: ACM, 2015), 1-21.
4. Bragdon, A., Nelson, E., Li, Y., & Hinckley, K., Experimental analysis of touch-screen gesture designs in mobile environments. *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems - CHI 2011,* (Vancouver, British Columbia, 2011), 403-412.
5. Jennings, A., Ryser, S., & Drews, F., Touch Screen Devices and the Effectiveness of User Interface Methods. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 57*(1), 2013, 1648-1652.
6. Katkoff, M., What the H _ _ _ Happened to Brawl Stars, https://www.deconstructoroffun.com/blog/2018/5/11/bs, (accessed 11 May 2018).
7. Li, Y., Beyond Pinch and Flick: Enriching Mobile Gesture Interaction. *Computer, 42*(12), 2009, 87-89.
8. Zaman, L., Natapov, D., & Teather, R. J., Touchscreens vs. traditional controllers in handheld gaming. *Proceedings of the International Academic Conference on the Future of Game Design and Technology - Futureplay 2010,* (Vancouver, British Columbia, 2010), 183-190.