

Design and Engineering of Software Applications for Touch Input Collection on Mobile Devices

Bogdan Gheran

MintViz Lab | MANSiD Research Center
University Ștefan cel Mare of Suceava, ROMANIA
E-mail: bogdan.gheran[at]gmail.com

Abstract

Recent advances in input technology have made touch screens the dominating interface for mobile devices and tablet computers. We developed in this work four applications that record users' touch input (taps and flicks) in both experimental and live settings. In this paper, we describe the design and development of these software tools. The variety of touch data reported by our tools (e.g., touch locations, timestamps, pressure, touch area, and device movement) will be useful to researchers to better understand users' touch input performance.

Keywords: Touch input, Gesture input, Mobile devices, Software tools, Android, Touch screens, Experiments, Touch measurements

1. Introduction

Gesture-based interfaces have had great impact on today's computing systems while smart devices allow human gestures to be captured in various forms. Previous studies on touch screen technology revealed that users prefer gesture commands that are different from those proposed by experienced designers (Morris et al., 2010); users are highly consistent with themselves when producing gestures (Anthony et al., 2013); users have various gesture preferences and still reasonable agreement levels can be established (Wobbrock et al., 2009); and that gesture execution in public depends on location and audience (Rico and Brewster, 2010).

Despite current efforts to recognize gestures on mobile devices and to understand how users articulate touch and stroke gestures (Anthony et al., 2013; Asbrook and Starner, 2010; Ruiz et al., 2011; Vatavu et al., 2013), we believe that the HCI community will benefit of more software tools to readily record users' gesture input in controlled experimental settings.

1.1 Paper Contributions

The tools that we present in this paper can be used to collect various gesture data, see below, and they offer the researcher a complete and unique dataset with gesture descriptions, composed of many touch related measurements. This data can be used as training for gesture recognizers or for evaluating gesture input performance directly. Our software tools will enable researchers to readily collect users' input on mobile devices.

2 Design of applications

All the applications were developed to meet the following requirements: quick time response, suitable processing algorithms, separation of gestures from involuntary touches. Our software tools run on Android and they collect touch gestures and other information regarding the device using the embedded accelerometer. All the data is recorded in XML files and consists of touch coordinates, area size of the fingers touching the screen, the pressure exerted by fingers on the

touch screen, the number of fingers used, the timestamps of the touch events, duration of the touch trials, and device orientation.

2.1 TapExperiment application

Targets are displayed on the screen in the form of circles that users must acquire as fast and accurately as possible. After each tap, the target changes location on the screen and also may change its size. The application is locked on the device's screen in order to prevent erroneous data recording by accidentally shutting down the device, tapping its hard and/or soft buttons or to prevent the display of third party messages on the status bar. We preselected the default size of targets following Google's recommendations (48×48 dp), because the average size of an adult thumb pad is about 10 mm.



Figure 1. TapExperiment configuration interface

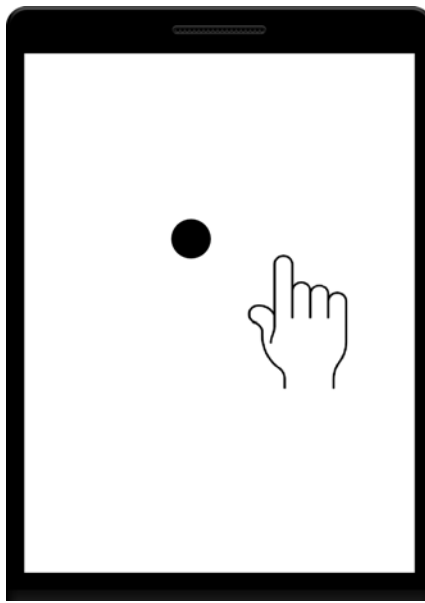


Figure 2. Testing TapExperiment application

The application was designed to be device independent considering that Android systems run on a variety of devices with various screen resolutions and form factors. Although the operating system resizes automatically the application, we made sure that the targets are displayed correctly according to the device *dpi*.

More complex touch experiments investigate how users can acquire targets displayed near the margins and in the screen corners, although the central area has the highest visibility and elements placed there are easier to touch. Therefore, approximately 90% of all targets are automatically placed in these challenging areas, while the rest are located in the center. The display modality is the same for all targets: black targets on a white background.

Once the application is launched, users can go through a training module or can start directly the experiment (see Figure 1). The application displays targets sequentially (see Figure 2). Users must acquire each target before proceeding to next one. A validation algorithm determines if the

touch point (captured when the user lifts the finger) is below a certain distance threshold from the center of the target. If the user fails acquiring the target accurately, they will receive a warning.

The TapExperiment application records the coordinates of the touch events, the pressure exerted by the finger on the touchscreen, device tilt, the task time and the touch time, the IDs of fingers touching the device, the medium acceleration of device on all the three axes, the size of the touched area, the target's radius, the offset between the target's center and the point where the user lifts his finger from the touch screen.

2.2 DensityTapExperiment application

The DensityTapExperiment (see Figures 3 and 4) collects users' touch input for scenarios where multiple targets are displayed close together, i.e., high density of targets.



Figure 3. DensityTapExperiment configuration interface

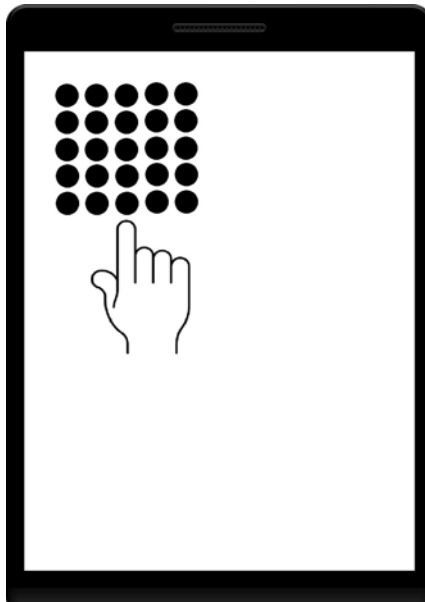


Figure 4. Testing DensityTapExperiment application

The application displays a matrix of 5×5 targets and the user's task is to select the target in the center (see Figure 4). The location of the matrix varies randomly on the screen, while the target's radius and the padding between targets are configurable.

The recorded parameters are: coordinates of the touch event, the pressure exerted by the finger on the touchscreen, device tilt, the ID of the fingers touching the device, the medium accelerations of device on all the three axes, size of the touched area, the target's radius, the padding value between targets, the offset between the target's center and the point where the user lifts his finger from touch screen. We also record time related parameters, such as the task time and touch time.

2.3 MovingTap application

In order to collect users' tap gestures for a moving target, we developed the MovingTap application. The application allows to set the target's radius and the movement speed from the configuration page (see Figure 5). By default, there are 5 values for target radius and 3 different speeds.

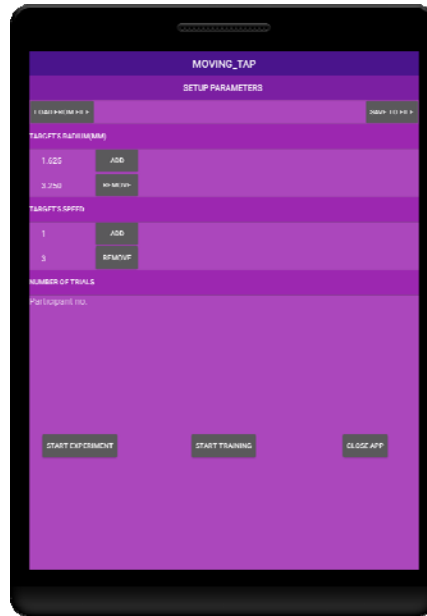


Figure 5. MovingTap configuration interface

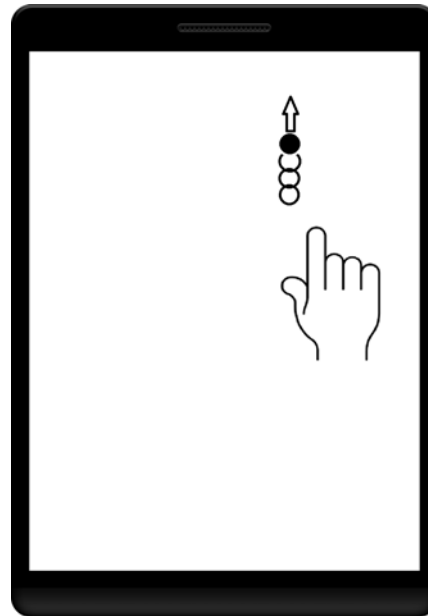


Figure 6. Testing MovingTap application

The application displays a target that is moving up and down or from left to right waiting for a user to touch the screen. The area where the target is placed is randomly changed for every trial. By default, there are implemented 3 different target velocities: slow, medium and high.

This software tool records the following parameters: coordinates of the touch event, pressure exerted by finger on the touchscreen, device tilt, the IDs of finger on the touchscreen, the medium acceleration of device on all three axes, the size of the touched area, the target's radius, the task time, the touch time, the offset between the target's center and the point where the user lifts his finger from touchscreen. We also record target related parameters like the radius and speed.

2.4 Drag and Drop application

Drag and drop is a technique for manipulating digital objects on a visual graphic interface. Very popular among PCs users, it has become a key element for touch screens as well. Therefore, we implemented a new application to collect users' drag and drop gestures. The measurements that the application records are: reaction time before the target is touched, the time needed to accomplish the task, the fingers' path from the start point to the destination and the number of fingers that touch the screen during this time, the distance between target and destination, pressure

and touch area size. We also record device-related parameters, such as accelerated motion and device tilt.

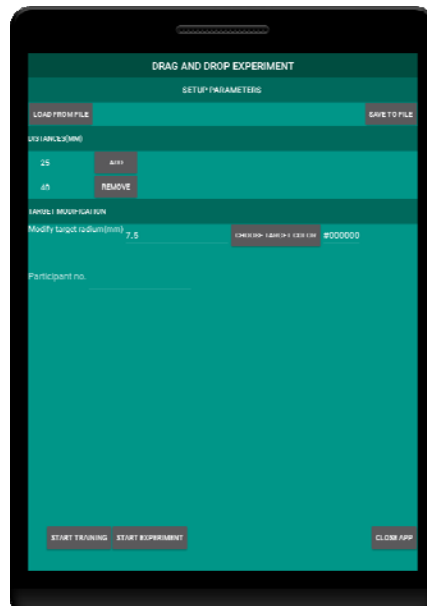


Figure 7. Drag and Drop configuration interface

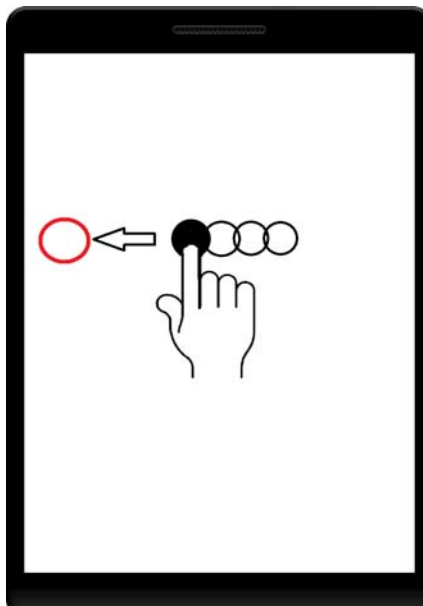


Figure 8. Testing Drag and Drop application

3 Conclusion

In this paper, we presented four software tools that collect users' touch gesture input. These tools are designed to run on Android devices. The variety of data and touch measurements collected will help researchers in their studies to design more efficient interfaces for users and especially for those with sight impairments. The software tools are free to use and can be obtained by contacting the author.

ACKNOWLEDGMENT

This work was supported from the project "Gesture interfaces for visually-impairing interaction contexts", no. PN-II-RU-TE-2014-4-1187 financed by UEFISCDI, Romania.

References

- Anthony Lisa, Vatavu Radu-Daniel, Wobbrock, Jacob O. (2013) *Understanding the consistency of users' pen and finger stroke gesture articulation*, GI'13, the 39th Graphics Interface Conference (Regina, Saskatchewan, Canada, May 2013). Toronto, Ontario: Canadian Information Processing Society, pp. 87-94
- Ashbrook D. and Starner T., (2010) *MAGIC: A motion gesture design tool*, SIGCHI Conference on Human Factors in Computing Systems, New York, NY, USA, 2010, pp. 2159–2168.
- Morris Meredith Ringel, Wobbrock Jacob O., and Wilson Andrew D. (2010): *Understanding users' preferences for surface gestures* Graphics Interface (GI '10). Canadian Information Processing Society, Toronto, Ont., Canada, 2010, pp. 261-268.

- Ruiz Jaime, Li Yang, and Lank Edward (2011): *User-defined motion gestures for mobile interaction*, SIGCHI Conference on Human Factors in Computing Systems (CHI '11). ACM, New York, NY, USA, 2011, pp. 197-206.
- Wobbrock Jacob O., Morris Meredith Ringel, and Wilson Andrew D. (2009): *User-defined gestures for surface computing*, SIGCHI Conference on Human Factors in Computing Systems (CHI '09). ACM, New York, NY, USA, 2009, pp. 1083-1092.
- Conference Proceedings:
- Rico Julie and Brewster Stephen. (2010): *Usable gestures for mobile interfaces: evaluating social acceptability*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10). ACM, New York, NY, USA, 887-896.
- Vatavu Radu-Daniel, Anthony Lisa, Wobbrock Jacob O. (2013) : *Relative accuracy measures for stroke gestures*, In Proceedings of ICM'I13, the 15th ACM International Conference on Multimodal Interaction (Sydney, Australia, Dec. 2013). New York: ACM Press, pp. 279-286
- <https://material.google.com/layout/metrics-keylines.html>