Documentation Octopcous

Martin Unterreiner, 119981

August 24, 2020

1 Introduction

Gestures for user interaction are predestined for mobile touchscreen devices like smartphones or tablet. On small devices it can be a frustrating experience for the users. It also can improve the collaboration for multi user application. Every user interactively follow the discussion. Oliver Bau wanted to improve the gesture-based interface and provided an example of a dynamic guided gesture recognizer. In his work he combined an on-screen feedforward and feedback system to allow dynamically gestures in different direction. The user can evoke different commands, also the application remembers the way, it was performed. Therefore in OctoPocus a menu is shown with a press and wait gesture. When the user starts the gesture the progress is continuously updated until he finish. In the next section I will describe the feedforward and the Feedback mechanism.

Feedforward The feedforward mechanism gives the user an impression of the shape of the gesture. It also describes the command it is related. Bau and Mackay uses predefined templates to represent the "gesture vocabulary". The starting points of all templates are the same and are initialized where the user tips the first time the screen with a "press and wait "gesture.

Feedback To provide the user a feedback, followed path is getting progressively thinner. With this gesture OctoPocus indicates that the application is recognizing is gesture. The progress of the gesture is calculated in realtime. Bau and Mackay adapted in OctoPocus a version of [2] algorithm.

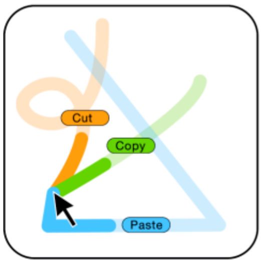


Figure 1: Feedforward.Bau and Mackay, 2008.

The gesture recognition is conducted by measuring the distance.

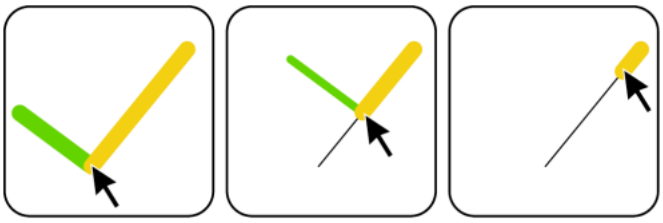


Figure 2: Feedback.Bau and Mackay, 2008.

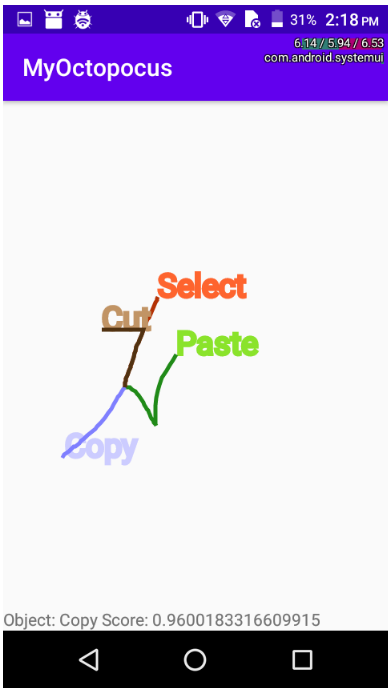
2 My implementation

In my first iteration I tried to implement a own implementation of Baus description within the dollar recognizer [1]. My first aim was to implement the menu and the recognizer with the main idea of feedforward and feedback. The second aim was to provide the execution of the command. Therefore I started to review the work of Juniior77 [2] and afterwards the MIS Project of Magdalena Keil and Jula McGibbon.

[1] https://github.com/olwal/dollar

[2] https://github.com/Juniior77/Octopocus

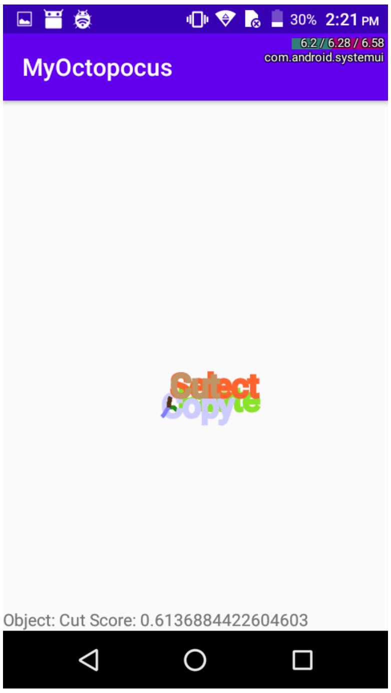
In the beginning I started with the drawing and the implementation of the gesture recognizing. But after having problems with the implementation of the recognizer of Alex Olwal I have fallen back to implementation of Juniior77, Magdalena Keil and Jula McGibbon. Most of my actual work of this implementation is some refactoring. My core idea of my implementation was to provide the user a menu which should disappear during movement of the gesture. My application doesn't difference between a Novice or Expert mode nor it doesn't learn the gestures. In fact I wanted to provide the menu (feedforward) even after a release of the finger tip to prove the user some time to memories the different paths.



Object: Copy Score: 0.9600183316609915

Figure 3: Menu with the feedforward path.

After starting the application is waiting for the first touch of the user. As soon as a onTouchEvent is recognized the menu with the feedforward paths is shown 2. The user can now follow one of the shown path. During the movement the menu is getting smaller. So during the action movement the actual position is saved. During the whole process the view within the painting is invalidated. After releasing the screen the recognizer is called. If he recognize a path from the given template paths, it will print the result in a textview. :w



Object: Cut Score: 0.6136884422604603

Figure 4: Menu with the decreased feedforward path.

References

[1] Olivier Bau and Wendy E. Mackay. "OctoPocus". In: *Proceedings of the 21st annual ACM symposium on User interface software and technology - UIST '08*. ACM Press, 2008. DOI: 10.1145/1449715.1449724.

[2] Dean Rubine. "Specifying gestures by example". In: *ACM SIGGRAPH Computer Graphics* 25.4 (July 1991), pp. 329–337. DOI: 10.1145/127719.122753.