ENTROPHY

Entrophy: user guide

David Marciel Pariente

davidmarciel@hotmail.com

Introduction

Entrophy is an application designed to increase security in password entry. It is inspired by the principles of the concept of its very same name, and it was presented at the University of Valladolid (School of Computer Science) by David Marciel Pariente as his Final Degree Project in September of 2015.

In this brief guide we aim to describe Entrophy in a quick and simple way by explaining its elements, but not the theory it is based on.

For its correct functioning it is recommendable to use mobile devices with high screen resolution (preferably 720x1280 or higher) and a minimum api of 17 (4.2.2). In case of not having a device that fulfills the requirements a virtual machine could be used.

OS Android greatly simplifies the installation of the application. All what is needed is to run the ".apk" file and follow the steps indicated by accepting possible permission requests.

Besides this, it is worth mentioning that before being able to use the application for the first time it is necessary to set a password.

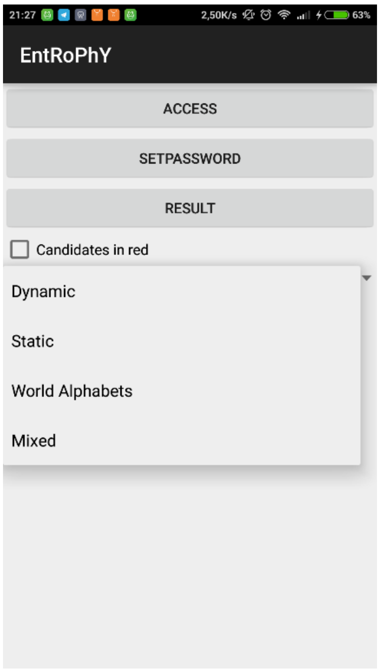
The application consists of four different views, each one devoted to a specific function. The views are:

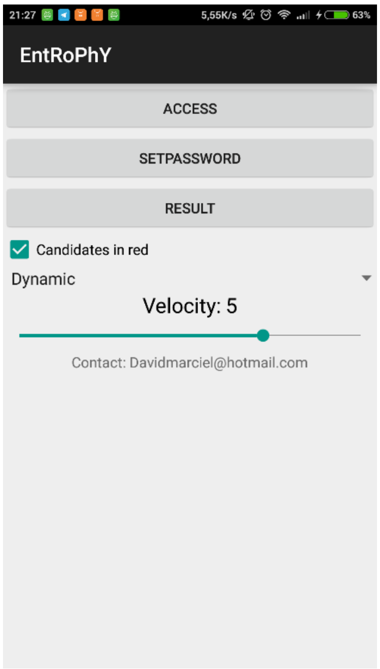
1. Launcher
2. Access
3. Set password
4. Result

1. Launcher

This view is in charge of the access to the others. It consists of:

- Three buttons that are to *launch* the other views (“Access”, “Set password” and “Result”)
- A checkbox that offers the possibility to choose the letters presented in the view.
- A mode selector. When setting password, it has to be taken into account that each mode has a different letter style.
- A bar that allows us to change letters’ the speed of motion.





☑ Candidates in red

Dynamic

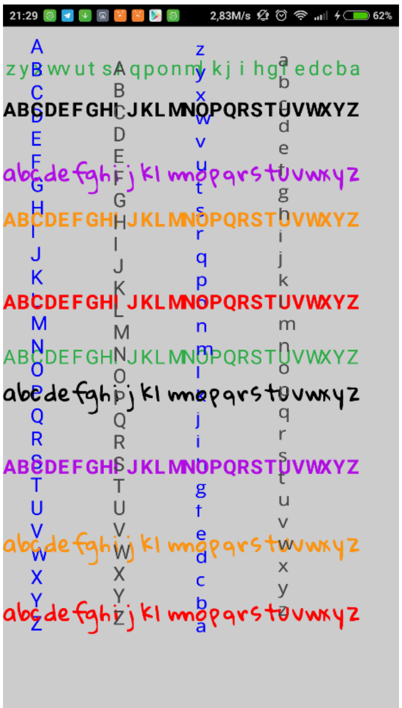
Velocity: 5

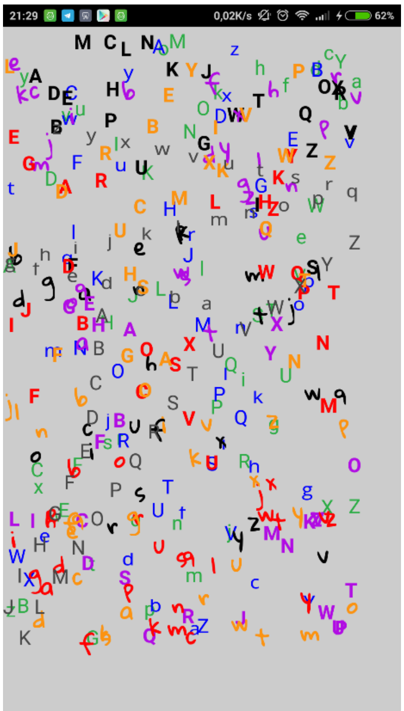
Contact: Davidmarciel@hotmail.com

2. Access

It is the main view, dedicated to show information concerning authentication.

It portrays a multitude of immobile letters which will be set in motion after touching the screen for the first time. From that moment, we can start picking letters.

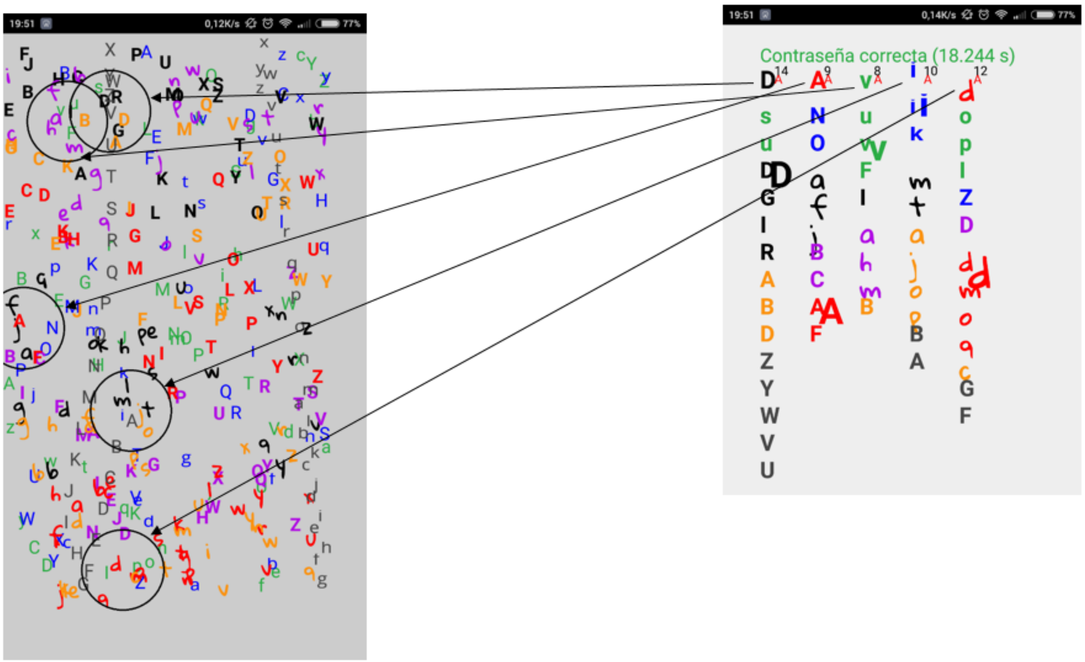




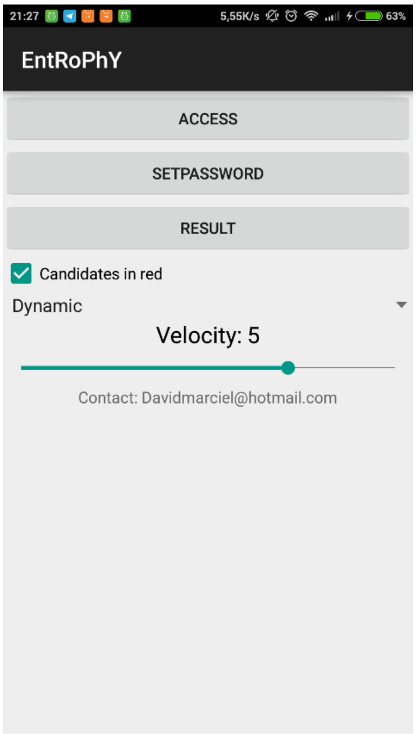
To prove that we know the password we have to pick all its letters in order. All the letters surrounding the touched point of the screen will be picked as "candidates" for the password: it is not necessary to touch the screen exactly where the wanted letter is. Doing it near the letter will be enough.

With this procedure it is impossible to know which, among all the "candidate" letters picked, is the correct one. Our touches on the screen will not reveal the password to an external observer because she will not be able to identify which letters –among all the candidates- we actually choose.

It is highly recommendable –at least for the first times- to localize the letters belonging to our password before touching the screen and set all the letters in motion. By the simple technique of knowing in advance where the letters we search are, we will later be able to find them more easily when they are moving. At the same time, this will contribute to protect ourselves from indiscreet observers by reducing our exposure time.


Contraseña correcta (18.244 s)

To see better which letters we pick, we can choose “Candidates in red”. This will highlight in red the groups of “candidate” letters when we are in “Access” view.

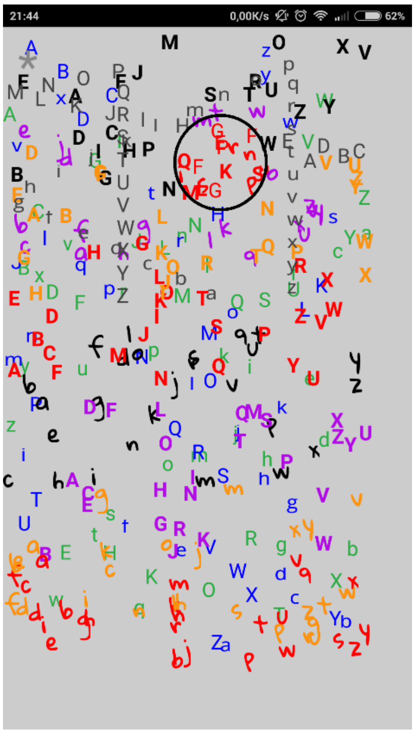


☑ Candidates in red

Dynamic

Velocity: 5

Contact: Davidmarciel@hotmail.com

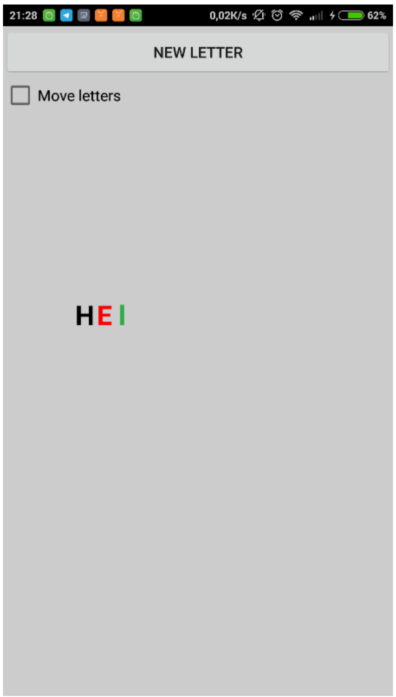


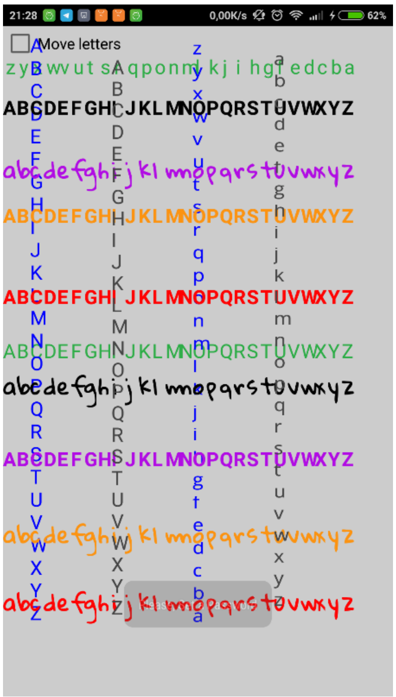
3. Set Password

“Set-password” view shows:

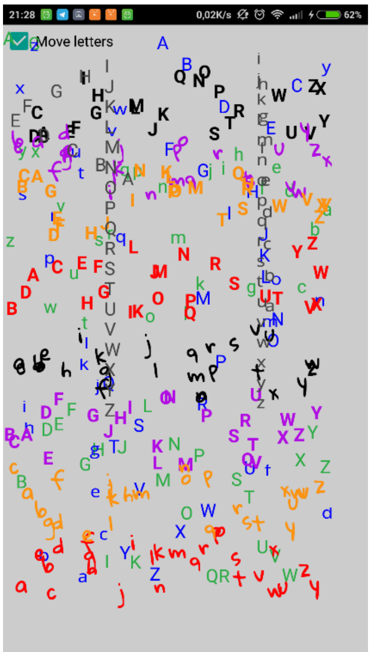
- Current password.
- A checkbox that allows blocking letters movement after picking them.
- A button that allows selecting a new letter for the password.

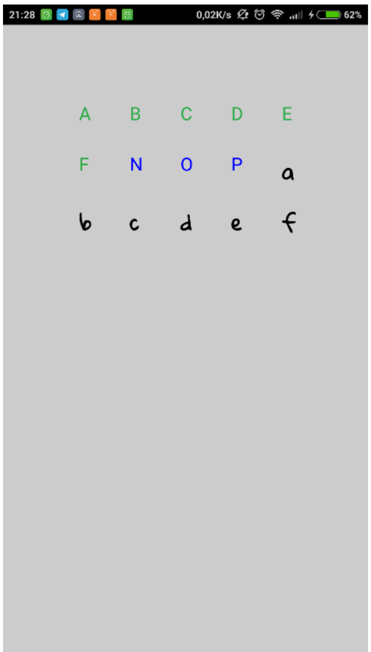
When we choose “New letter”, a view similar to “Access” is shown. This view has a checkbox that allows us to stop and resume letters’ movement.





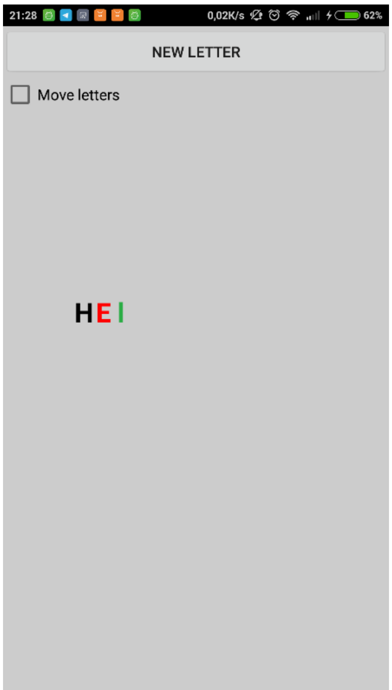
After touching the screen, a logarithmic chooser shows the “candidate” letters (those closer to the point touched on the screen), so that we can easily select the one we want to be part of the new password.

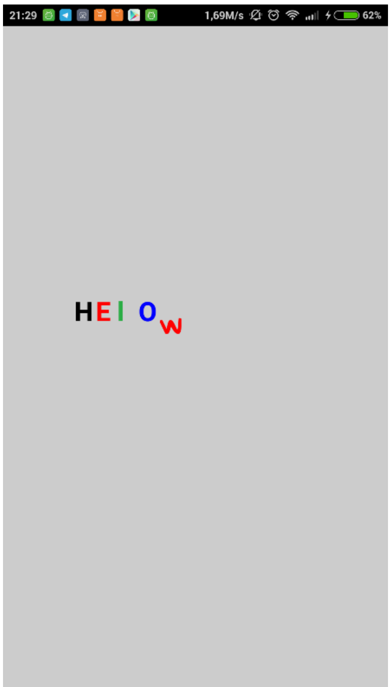




Once the wanted letter has been chosen it will be shown alone and then we will be able to go on choosing letters by pushing on “New letter”. When the five letters for our password have been chosen, the new password will be set and the button to add new letters will not appear again.

From this moment on the new password will be used by the “Result” view to check access.

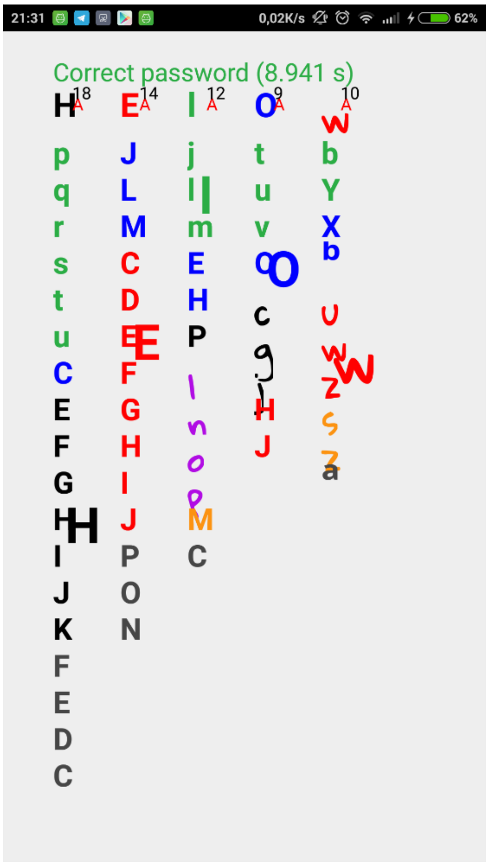




4. Result

The “Result” view is accessible from the launcher. It shows information about the last access attempt. This view presents

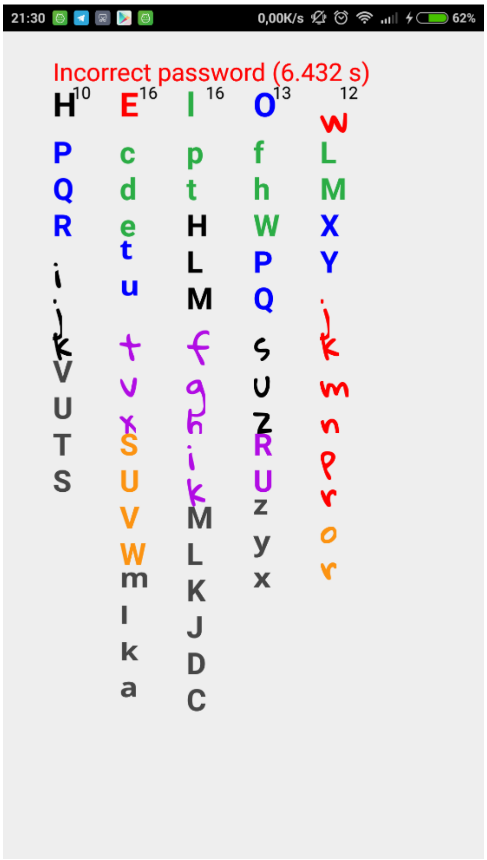
- Firstly, a message saying whether the attempt was successful (“Correct password” or “Incorrect password”) and how long (in seconds) it took.
- Under that information, the current password appears in big letters. Each letter of the password has a superscript figure showing the number of “candidate” letters obtained in the last access attempt. If the correct letter were in that set of “candidate” letters, it would also have a red “A” (meaning “accepted”) as a subscript.
- The sets of “candidate” letters –each one created by one screen touch- are unfolded as columns under the corresponding letter of the password. If it the set contains the correct letter, this will be replicated at its side in a bigger size.



21:31

0,02K/s 62%

Correct password (8.941 s)



21:30

0,00K/s 62%

Incorrect password (6.432 s)

The “Result” view is only a debugging view which would not appear in a final application. In the final application this view would not be shown because information related to the password would be kept by the server and never shared with the client nor other third parties.

In fact, the device will never get to know the password. It will simply act as an information transmitter between user and server, being the latter the responsible for validating whether the position and the time of screen touches correspond to the position and the time in which the letters of the password were.

This is how we obtain security even when our communication is known. Given that knowing the communication does not imply to know the password, an external observer could only know that the first letter of the password is within the first set of “candidate” letters, that the second letter of the password is within the second set, and so forth.

The last pictures depict an accepted password. The first set of “candidate” letters contains 18 letters, the second 14, the third 12, the fourth 9, and the fifth has 10 letters. This implies that the number of possible combinations is 18x14x12x9x10 = 272,160. Among all them there is only one valid (“HElOw”, in the correct colors), so that even if someone gets to know the communication, she will not be able to know which among those 272,160 possible combinations is the correct password.

Besides, given the pseudo-random nature of the letters’ motion, it is impossible to repeat the same set of letters and observers cannot therefore repeat their observations. This makes the system brute-force resistant.