Documentation ToothTest

Francisco Carrillo Pérez <franciscocp@correo.ugr.es>

03-09-2016

Abstract

This is going to be the documentation for the gingiva test program, that allows you to show different samples to an user, to test the color.

1 Prerequisites

You will need to have installed Python 3. In this link, Python you'll find the latest Python's version. You could install it in every OS and use it without any problem.

Also you'll need the library Tkinter, if it is not install in your Python version. You could find more information of the library and the installation here: TKinter.

2 What I'm going to find?

Here you are going to find 2 things:

- Samples Folder: Here is where you have to put the samples images. Also you can find here a Python's script to resize all the images. This script only works in Linux.
- toothtest.py: This is the main program. We'll learn more about it later.

3 How it works?

You have to run the gingiva.py script. There are different ways of doing it:

- In Windows you could just double click it and then it would run.
- In Unix based system, you could use your terminal and type: python3 toothtest.py

After using it, you will find that a .txt file have been created with all the choices that your observer have made.

4 toothtest.py

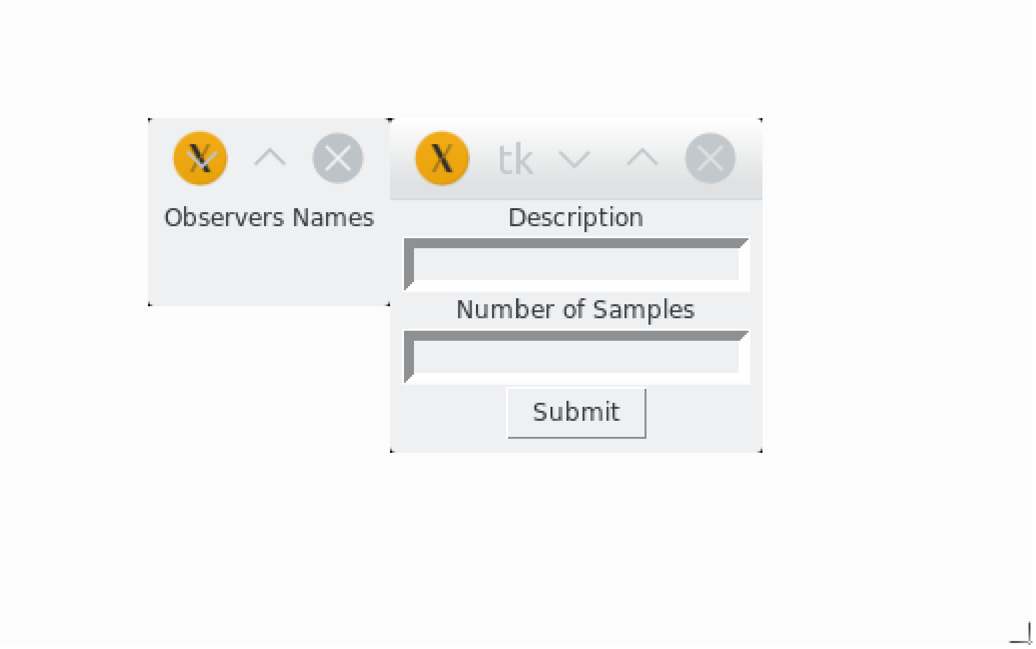
In the program you are going to find different parts. The only parts you must be concern about, are the lines where you find the words: **screen_width** and **screen_height**, for example:

```
# set first image on canvas
self.image_on_canvas = self.canvas.create_image((screen_width/2)-100,
screen_height/2, anchor = NW,
image = self.my_images[self.my_image_number])
```

This is used to set the position in the screen. It detect your screen's size and assign the width to screen_width and the height to screen_height. You can figure out that it's divided by 2 because we want it in the middle of the screen. So, you can change every number in the Python's script to adjust perfectly to you screen's size.

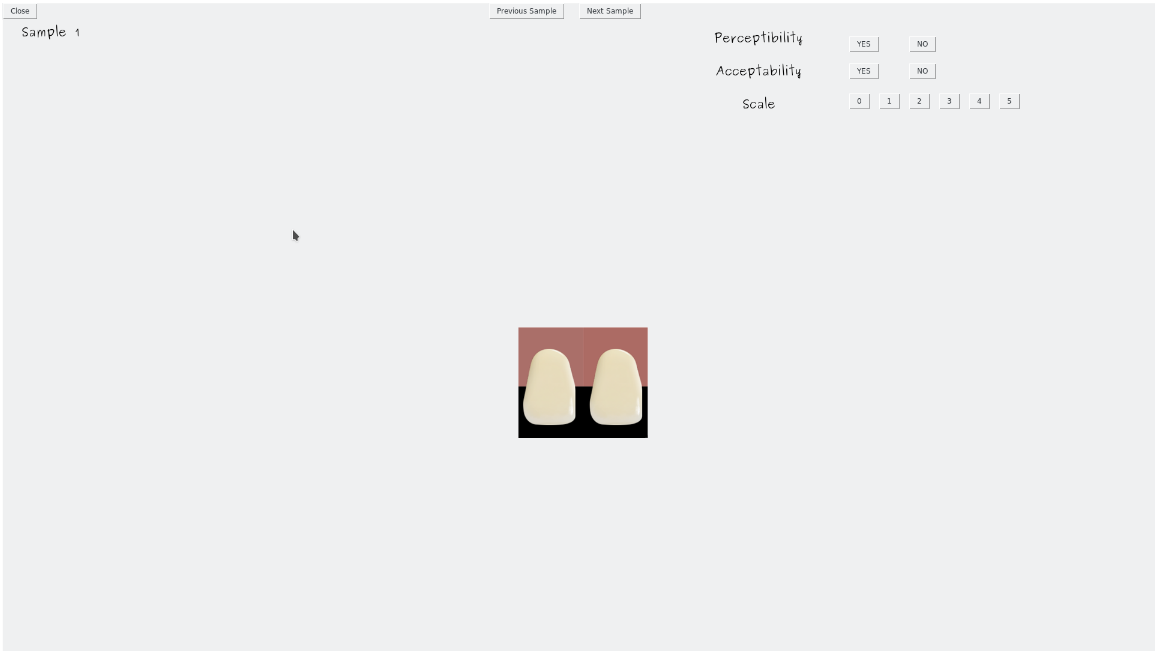
5 The UI

First what you're going to see is this window:



Here you can write the description of the observer and the number of samples that you're going to use.Also, you'll see the observers names that have already passed the test. This is going to be used to create the file were the results are going to be written in.

Then you'll see the main window:



So, you can see you have different buttons. Let's talk a little bit about them:

- **Close:** it close the window, as it name says.
- **Previous/Next Sample:** it change to the next or previous image.
- **Perceptibility:** If you can perceive color differences hit **YES**. Otherwise hit **NO** and you can change the sample.
- **Acceptability:** If you accept the color differences that you perceive hit **YES.** Otherwise hit **NO**. Keep going with the scale.
- **Scale:** Choose one number between 0 and 5. 0 is that you don't percieve any difference and 5 if they are very different. Then you can change the sample.

In your left side of the screen you have the sample's number.

6 Which files are going to be created?

You'll find that several files and folders are going to be created:

- **Observers Evaluation:** Here is where the results are goint to be saved. All of them have the name: observer"number of the observer".txt .
- **observers.txt:** Here ir where the names of the observers who have already passed the test are going to be stored.

7 Variables that I need to change in the code

This is a very important part of the documentation. You need to set the number of samples that are going to be used as samples, of course. You can change it in this line of code:

```
for i in range(1,number_of_samples):
    self.my_images.append(PhotoImage(file = "Samples/samples"+str(i)+".png"))
```

Here you can change :

- **The name of the samples:** all the samples **need to have the same name follow a number**, which goes from 1 to the number of samples you have.