

2. Our password scheme is called Pictanator, it's a picture based password. There are 7 categories of pictures, each category has 16 pictures/elements each as indicated below:

Categories with their Elements

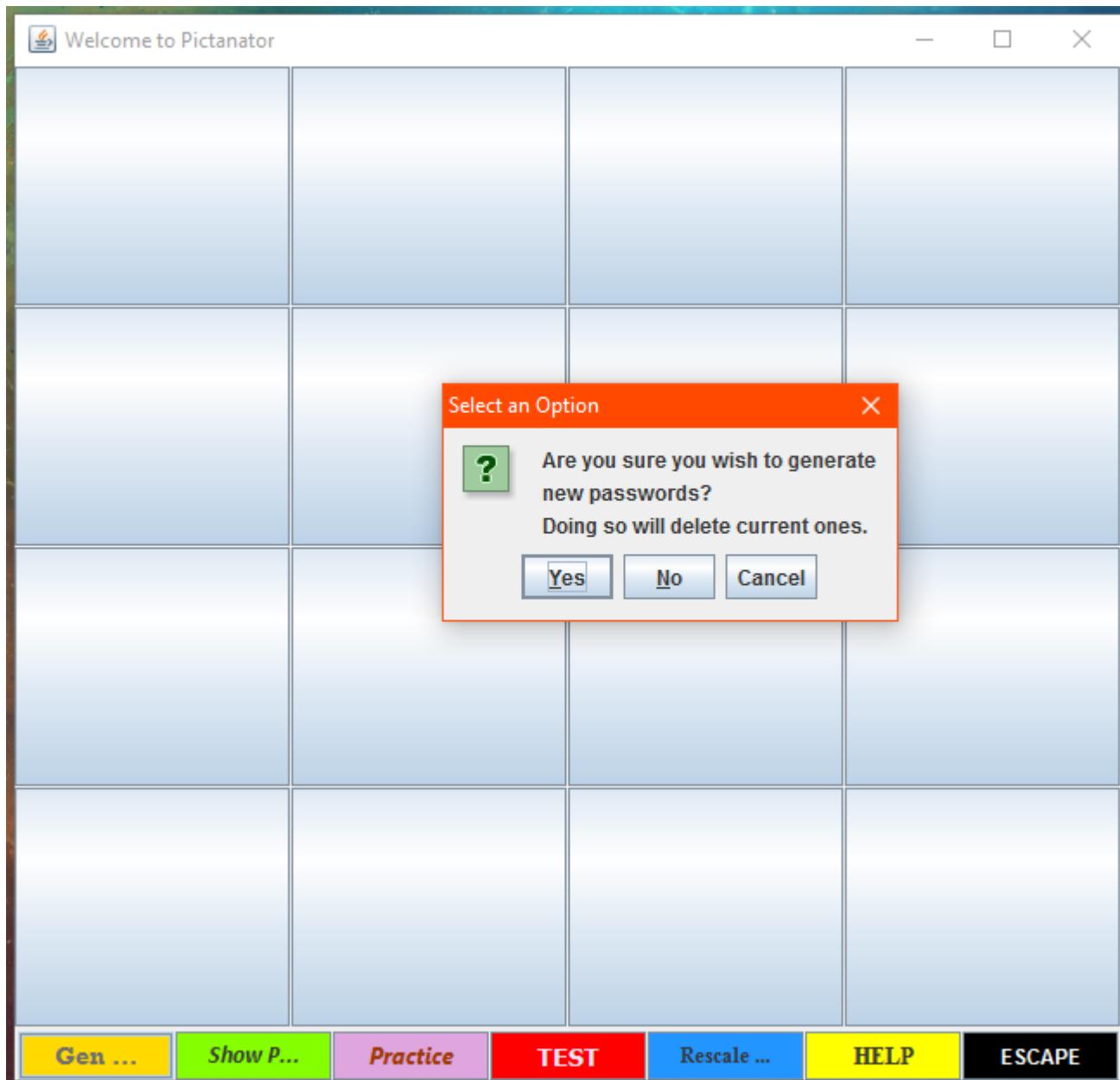
Index	Animals	Foods	Vehicles	Places	Sports	Clothing	Stuff
0	Kangaroo	Sushi	Submarine	Niagara Falls	Soccer	Hats	Gold
1	Frog	Pizza	Bicycle	Ocean Reef	Hockey	Pants	Television
2	Flamingo	Fries	Car	Pyramid	Basketball	Shoes	Computers
3	Octopus	Tacos	Airplane	Jungle	Baseball	Shirt	Meteorite
4	Lizard	Soup	Segway	Glacier	Tennis	Backpack	Crown
5	Beaver	Burger	Flying Saucer	Moon	Football	Sun Glasses	Fossil
6	Penguin	Salad	Boat	Great Wall	Cricket	Gloves	Sword
7	Elephant	Ice Cream	Helicopter	Volcano	Bowling	Socks	Telescope
8	Bear	Hot Dogs	Motorcycle	Island	Curling	Jackets	Jet-pack
9	Eagle	Lasagna	Hot Air Balloon	Tower	Ultimate Frisbee	Scarf	Lantern
10	Lion	Rice	Golf-cart	Beach	Golf	Space Suit	Talisman
11	Wolf	Waffles	Train	Cave	Boxing	Cape	Gyroscope
12	Giraffe	Shawarma	Truck	Castle	Archery	Dress	Pottery
13	Rhinoceros	Chicken Nuggets	Canoe	Savannah	Volleyball	Fire suit	Totem Pole
14	Shark	Omelette	Skateboard	Space Station	Rowing	Swimsuit	Bong
15	Dolphin	Pie	Snowmobile	Mountain Top	Water Polo	Tuxedo	Treasure Map

Basically the user will have to memorize 1 element from each categories, therefore 7 total distinct images per password, for example, Dolphin, Pie, Snowmobile, Mountain Top, Water Polo, Tuxedo and Treasure Map.

The following is a brief overview, detailed information will follow. The way this program works is first it sets up the GUI objects, so the panels, buttons, icons, etc. There are currently 2 panels, a panel for 16 buttons and a panel for action based buttons. There are 112 pictures that are loaded into the application which are PNG type files to ensure top quality when resizing. All the pictures and their names are loaded and initialized. There is another window created that displays the password selected. The first panel that contains 16 buttons has pictures overlay-ed on them, so the user can simply click on their password, the second panel contains 7 button that contain functionality when clicked such as password generation to help. This section, question 2, will talk about all the functionality of the program and question 3 will talk about the testing functionality.

Gen Pass

The Gen Pass button stands for generate passwords, this button will ask for confirmation and then delete overwrite any 3 existing passwords with newly randomly generated passwords.



Show Pass

The show pass button simply asks you which password you would like to see, either Password 1, 2, 3 and then displays it in a new window.

>Welcome to Pictanator

Show Password Selection

Please choose a password to see.
Password 1

OK Cancel

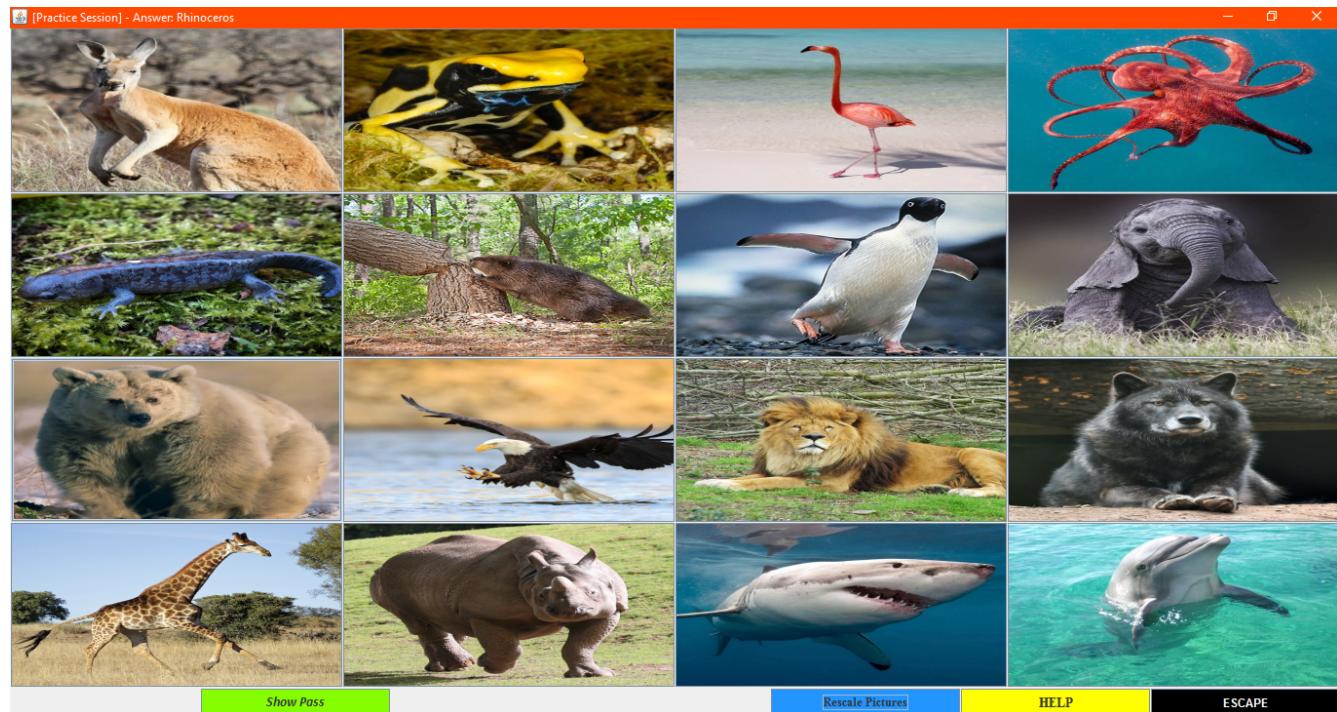
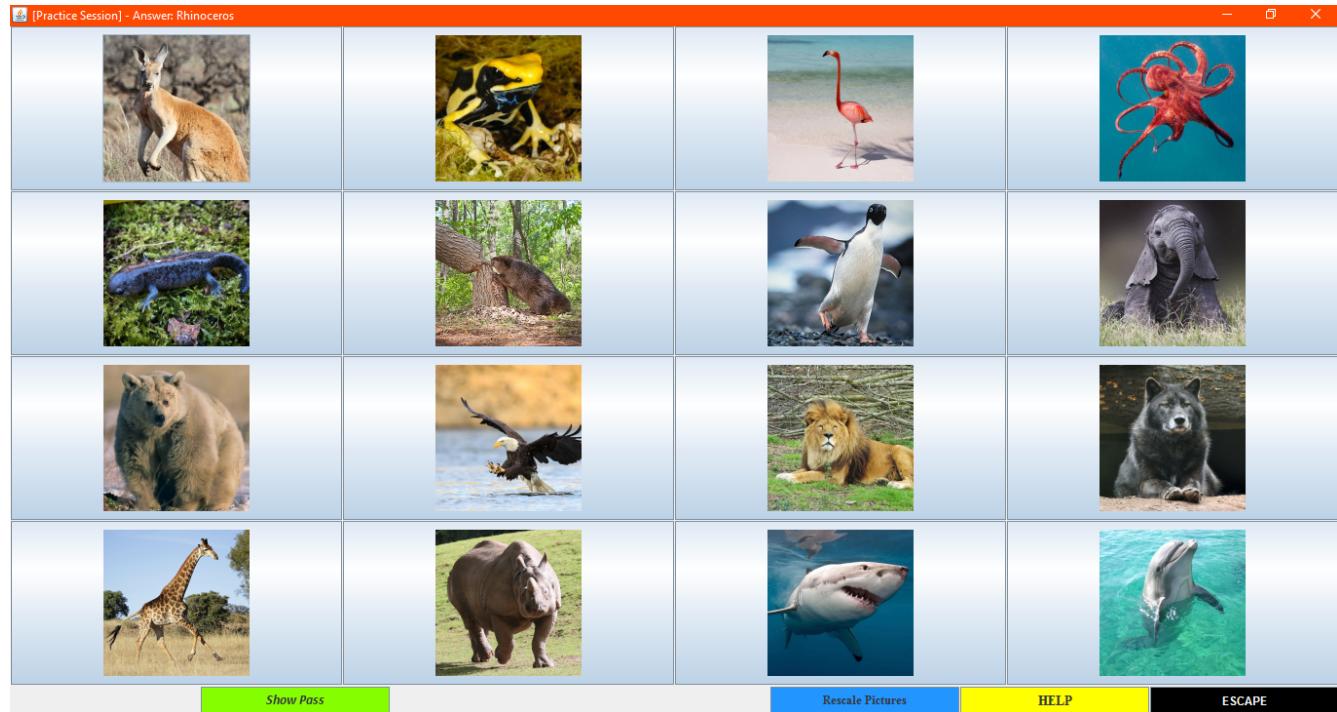
Gen ... Show P... Practice TEST Rescale ... HELP ESCAPE

Your Password

Animals	Foods	Vehicles	Places
Rhinoceros	Pie	Car	Island
			
Sports	Clothing	Stuff	Summary
Hockey	Shoes	Meteorite	Rhinoceros Pie Car Island Hockey Shoes Meteorite
			

Rescale Pictures

This button will re-scale the pictures if the window is resized and the pictures are too small for the user. Although the pictures are all loaded with dimensions of 150x150 some users may be using screen with a larger resolution, therefore ensuring accessibility is an important aspect of GUI design. The following pictures show a Before & After images.

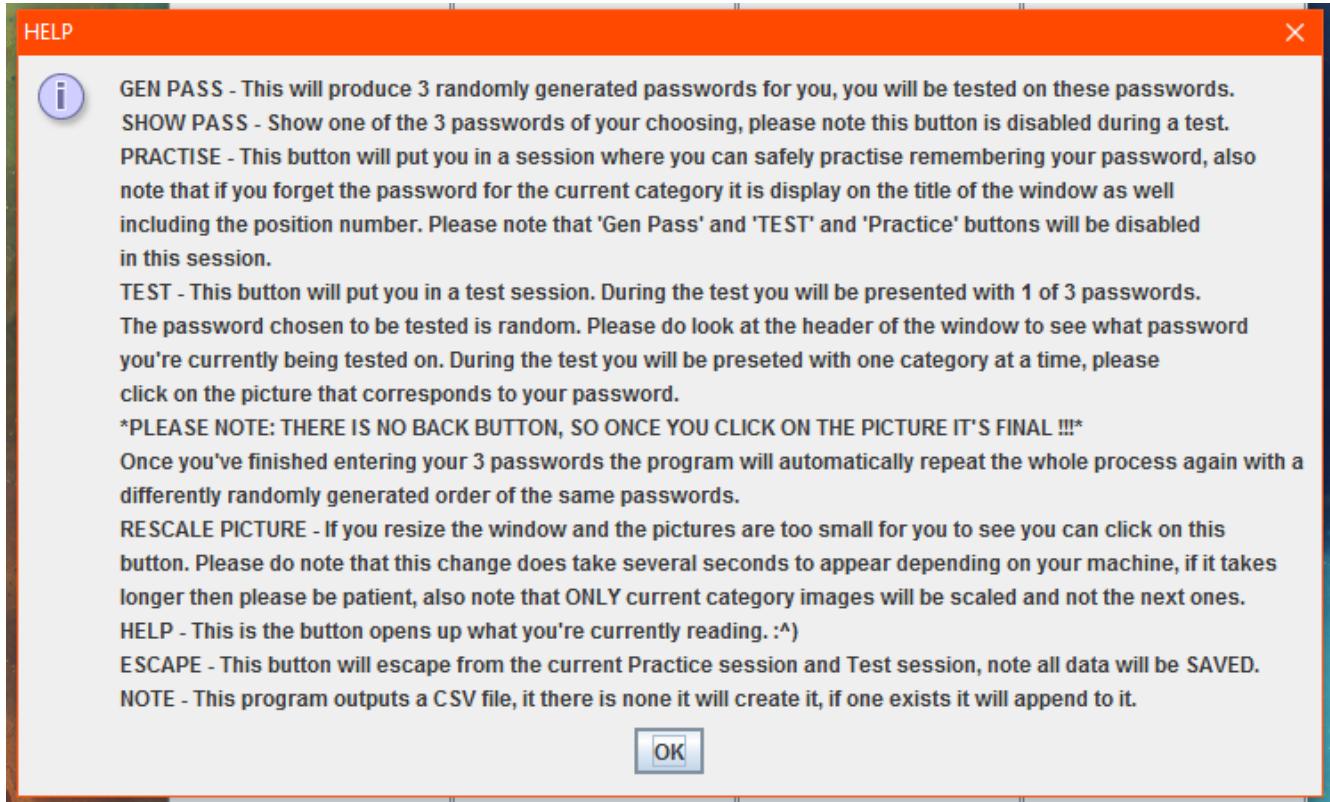


ESCAPE

The escape button will simply exit out of the current state of the program, so if you are practicing or taking a test this will allow you to exit right away, the results are currently always saved whether you press the escape button or not, however this was made to simplify testing and easier usability for the TA. Once pressed it resets the state of the program and all the settings return to normal.

HELP

All the instructions and information are a click away from this button. This will display all the important information on all the buttons, how the practice sessions and test sessions work and any other important information. Below is a snapshot of how this looks:



Structure of the Pictanator

This program responds and changes from input from the user by altering and modifying its state, for example if passwords are generated they are ready to be deployed, if the show password is clicked the program is ready to display the text and pictures. If testing of this program is to be done in detail it is highly advised to check it by loading in the program in an Eclipse environment and running the program while having the console open, the `System.out.println()` have been done in great detail to ensure simple testing at all times during testing and practice. Re-scaling the pictures is done on the fly, meaning whenever you press the "Rescale Pictures" button the program must calculate and redraw the images, this is a more appropriate approach since it's dynamic and allows the program to adjust accordingly to the screen at hand, especially since this won't always be used.

To elaborate in-more detail about how the program runs is that its architecture is asynchronous and this is very important because imagine if this program had while loops with many if statement, if any changes are made it is likely to crash, even while running it is likely to crash. Instead the program

simply reacts when a button is clicked and changes it's state, then another button is pressed, variables and states are changed, then another button is pressed.....etc. There are a few benefits to it, first it consumes almost no CPU resources, if no button is pressed then simply nothing is happening. Second is that it's extremely fast and responsive because it has a fixed set of instructions to carry out whenever a button is clicked, therefore creates a better user experience since the program never "lags" and never causes to "slow down" the computer. Third is that it influences smart designs to focus on efficiency and usability, it's efficient because it only uses the resources when it needs to do work and it's usable because it's fast and doesn't affect the whole computer when used.

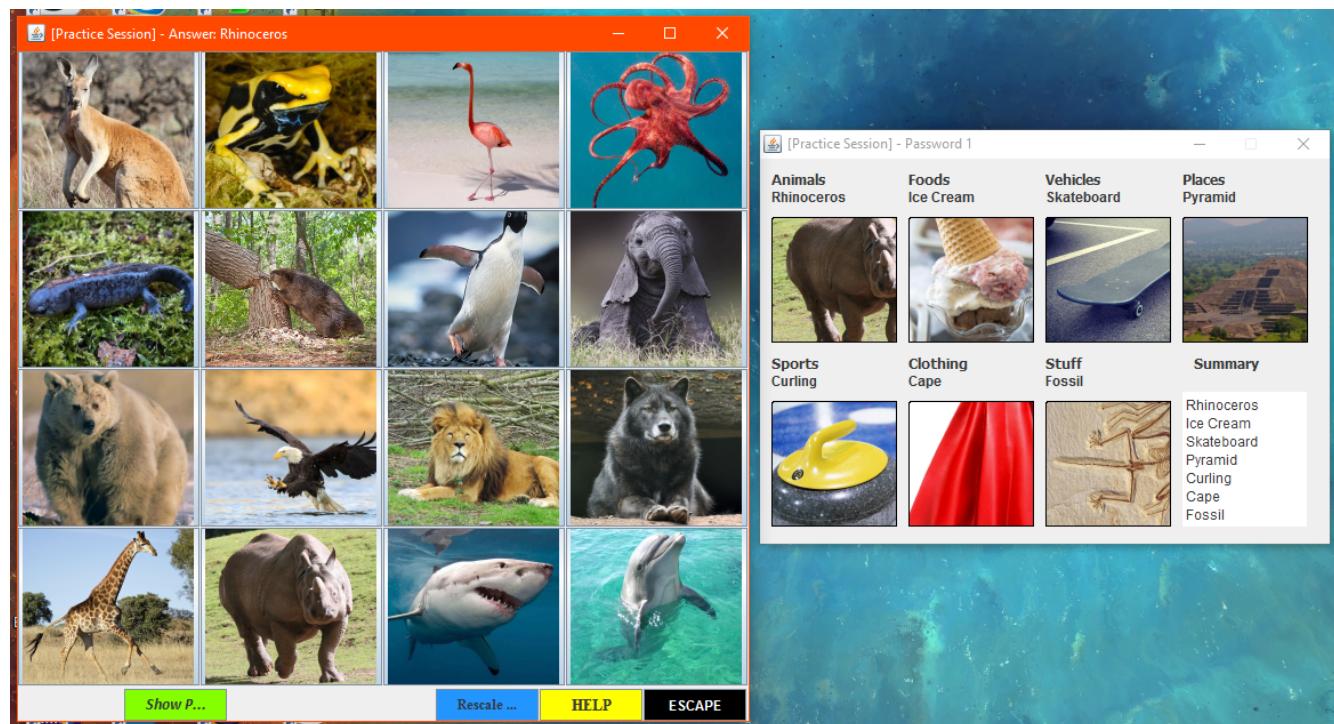
Another important note to mention is that the **ActionListener** also helps in keeping the program asynchronous. The program also takes best practices into account such as encapsulation, code-reuse set as functions, etc.

However there are cons to this design approach. First the load time takes to long, when this program becomes a jar it has to load the pictures from the resources folder within itself to memory and process all 112 images, set and initialize the objects, etc, therefore compile time will take at least 10-20 second, it really depends on you CPU, but this makes run time a breeze. Another con is that if a change needs to be done it makes it very difficult because before applying your change to the code you must take into account all the values and objects that will be affected by this change during initialization, compile time, run time. Everything must be handled and you cannot have mistakes or you will suffer greatly.

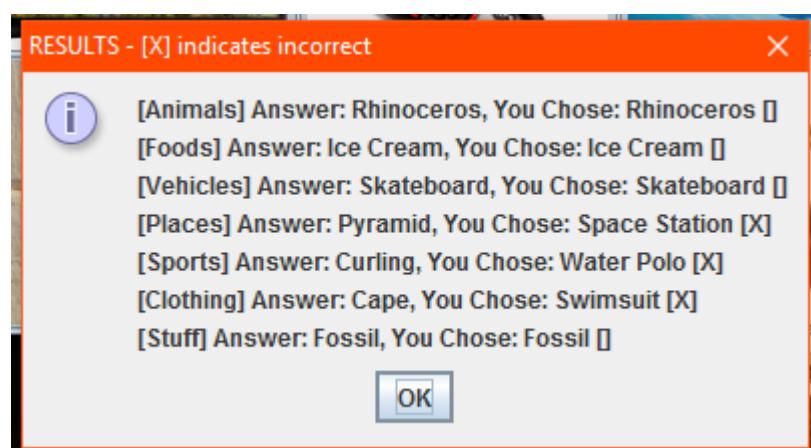
3. The framework is actually built inside the program, just like other functions generate passwords, Practice and Test buttons also have an asynchronous structure with state values, the program also exports a CSV file by either creating one or appending to one with the same name and extension, more detail will follow.

Practice

The practice button will pop up and ask you to choose either Password 1, Password 2 or Password 3. Once that's images will appear and a new window will pop up, as show below:



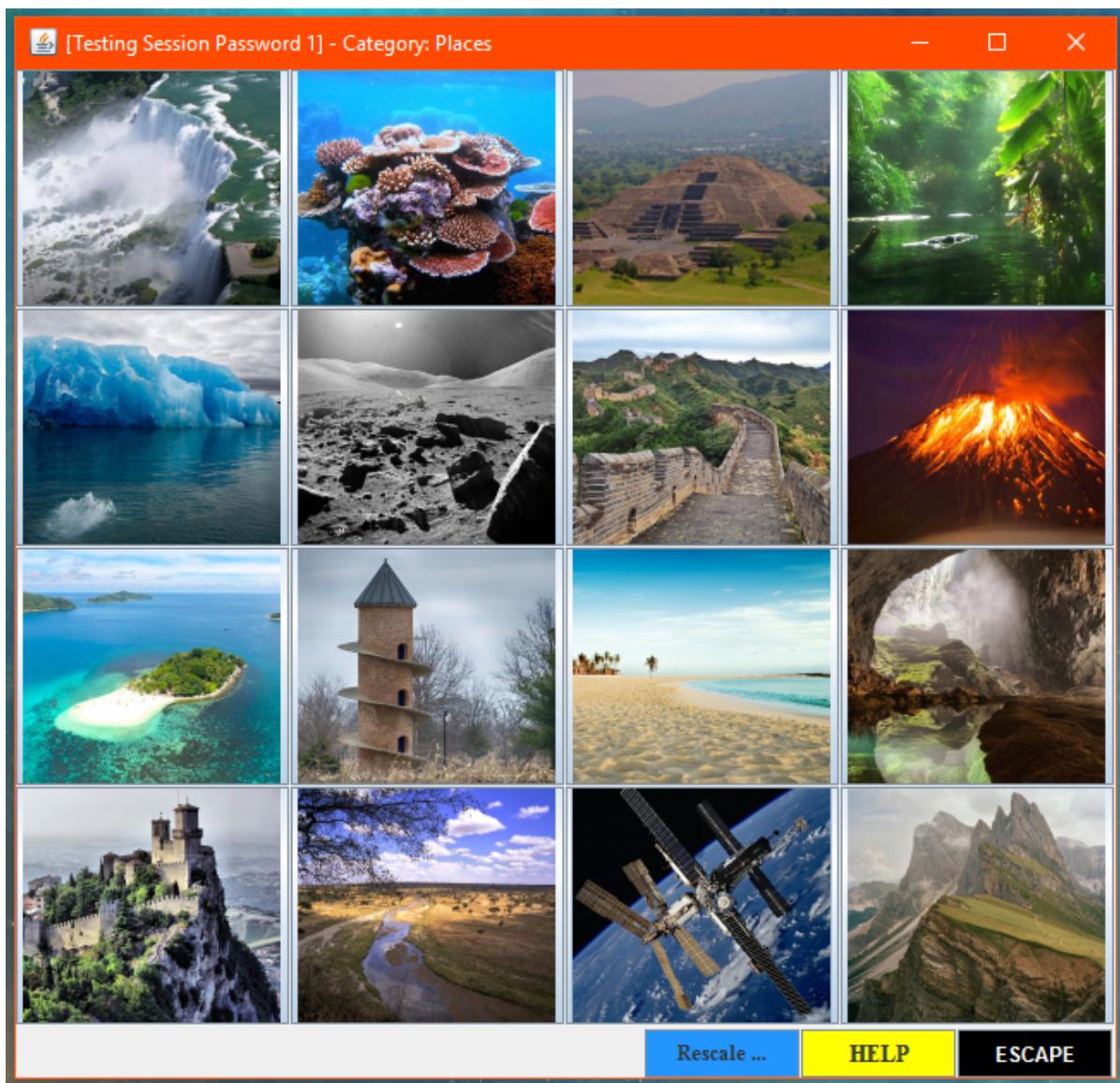
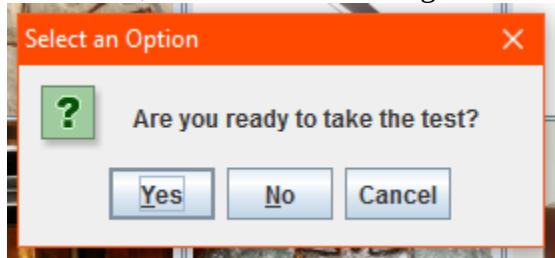
During a practice session some buttons have been disabled to ensure there is no change to the session. Once the password has been entered a new pop up information window will display the results, below is an example of the above continued:



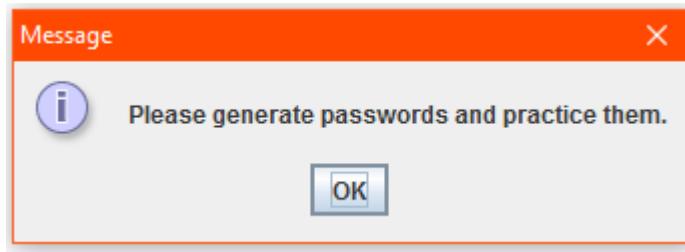
Once you click OK click on the Practice button again and practice another password.

TEST

When the user is ready to take the test click on the TEST button to get started. Click Yes on the confirmation screen, once clicked the test will start immediately, so the program will randomly order the 3 passwords and display them in the random order created. For example, 2,1,3 is the order in which you will see the passwords, also looking above at the top shows which password is currently being tested and which category is awaiting its respective correct input. There are 3 chances to enter passwords, after 3 chances are used up the program moves on to the next password, the program will give you a pop up message window indicating that it is switching to the next password. Below is an image that shows the warning confirmation window, then a window during the test:



Note - If the user tries to enter a practice session or test without generating a passwords first they will receive this message:



This will prevent any “accidents” for the program or the user.

Bug

Luckily there is only one bug. The instructions of the project was to create 3 random passwords, and test each of them in a random order, then test them again in another random order. For example [2,1,3] and then [3,2,1]. The program during testing was crashing and having issues while attempting to get 3 random passwords to work twice while half-way randomizing the order again. However it is very stable now but doesn't meet the specifications exactly, in other words it never goes past 5 testing rounds. It will do for example something like this [2,1] & [3,1,2]. In the order of 2 the first phase and 3 the second, we have attempted to fix it, but decided its better to leave it working since it's not crashing and we are able to pool in data no matter what, and meeting 5 randomly testing orders is better than none.

CSV File Format

The program outputs a file called “dataOutput.csv” and it’s in the following format:
“UserID,CurrentPassword,Category,Success/Failure,Answer,CorrectAnswer,TimeSpent”

UserID – The unique user identifier for each test participant, automatically incremented

CurrentPassword – The current password being test, i.e. Password 1

Category – The category on which the program is currently testing

Success/Failure – Either a the correct answer was chosen or not

Answer – The answer the user submitted

CorrectAnswer – The actual correct answers

TimeSpent – Time spent on the current category in seconds

Note: When the password has been fully answered, regardless if it’s correct or not, is also logged and takes the same format as above, however there is a difference, below is a sample.:

“1001,Password 2,Total/Overall,Failure,N/A,N/A,29.076”

Notice how the column fields answer and correctAnswer have values of “N/A” since they cannot carry an answer. This record of data also gives more insight on how the users are doing overall.