Project – Chess GUI with Engine  
Testing  
Harley Reeves-Martin

Contents

[3.0.0 Test Plan: 4](#_Toc8489419)

[3.1.0 Validation: 5](#_Toc8489420)

[3.2.0 Functionality of Processes: 6](#_Toc8489421)

[3.3.0 Data storage & manipulation: 7](#_Toc8489422)

[3.4.0 Linkage: 8](#_Toc8489423)

[3.5.0 Overall System: 9](#_Toc8489424)

[3.6.0 Proof: 10](#_Toc8489425)

# References

No references are made in this document.

3.0.0 Test Plan:

|  |  |
| --- | --- |
| Test series: | Description of test: |
| 1. Validation | This validates that wherever the user can make an input, if their input is valid it will be used as expected, and if it is not then the program will handle that appropriately. |
| 1. Functionality of Processes | This validates that the program behaves as expected by comparing program results to expected results. |
| 1. Data storage and manipulation | This validates that any files saved by the program are stored in the correct format and can be loaded again properly. |
| 1. Linkage | This validates that the program will never get stuck in one state. |
| 1. Overall system | This validates that the program does not crash, and no exceptions are thrown at any point. |

3.1.0 Validation:

| Test No. | What is being tested | Data | Expected Result | Actual Result | Evidence Reference |
| --- | --- | --- | --- | --- | --- |
| 1.1 | The user should only be able to enter strings between 4 and 20 characters long for a username on signup. | Valid: harley24  Extreme: ottf, ottffssentettffssent  Erroneous: kd2, zxcvbnmasdfghjklqwert | Valid and extreme data should both be accepted, and the form should close (indicating that the account has been created). Erroneous data should lead to an error message box being displayed. | As expected. | Evidence 1. |
| 1.2 | The user should only be able to enter strings between 5 and 20 characters long for a password on signup. | Valid: password123  Extreme:qwert, asdfghjklzxcvbnmqwer  Erroneous: ondf, thispasswordistoolongtohandle | Valid and extreme data should both be accepted, and the form should close (indicating that the account has been created). Erroneous data should lead to an error message box being displayed. | As expected. | Evidence 2. |
| 1.3 | The user should only be able to select PNG and JPG files to loaded as their account image. | Valid: any JPEG or PNG  Erroneous: Any file that is not JPEG or PNG, E.g. PDF. | Valid data should be accepted and should be shown as an image in the box, if an invalid file is chosen then a message box saying the image cannot be loaded should be displayed. | As expected | Evidence 3. |
| 1.4 | The user should only be able to create an account if the password and confirm password are the same. | Valid: password, password  Invalid: password, password123 | Valid data should be accepted, invalid data should cause an error message to be displayed and the account not created. | As expected. | Evidence 1, Evidence 4 |
| 1.5 | The user should only be able to enter an ELO in the range 800-3500 | Valid: 1950  Extreme: 800, 3500  Erroneous: 24000 | If the user enters a value greater than the max, then the numeric up down should change that value to 3500, and if they enter a value lower than 800, the NUD should change it to 800. | As expected | Evidence 1. |
| 1.6 | The user should only be able to enter RGB colours for the light and dark square in settings. | Valid: Any colour with an alpha greater than 0 | The choose colour dialog should only allow the user to input a colour. | As expected. | Evidence 5 |
| 1.7 | The user should only be able to select yes or no for the following settings: show legal, cheats and night mode. | Valid: true/false | The checkboxes only allow for Boolean inputs. | As expected | Evidence 6. |
| 1.8 | The user should only be able to input valid moves. | In the starting position:  Valid: e4, e3, d4, d3…  Invalid: Qe4, Rh8… | The GUI should update the board position if the move was legal if not, it should return to the original position. | As expected. | Evidence 7. |

3.2.0 Functionality of Processes:

| Test No. | What is being tested | Data | Expected Result | Actual Result | Evidence Reference |
| --- | --- | --- | --- | --- | --- |
| 2.1 | The board should only generate legal moves. | Starting position, looking at the e2 pawn. Looking at the g1 knight. | The pawn should be able to play the moves: e3, e4. The knight should be able to play the moves Nf3, Nh3. | As expected. | Evidence 8. |
| 2.2 | The user should be logged in and an account window shown when the user logs in. | Account details for a valid account (username: harley15, password: \*\*\*\*\*\*\*\*\*) | The window should be shown. | As expected. | Evidence 9. |
| 2.3 | The move list should update itself accordingly when a move is played. | From starting position, when c4, e5 is played, the move list should show that. | The move list should show (1. c4, c5) | As expected. | Evidence 7. |
| 2.4 | The move list should update its opening text accordingly when a move is played. | From starting position, when e4… is played, the text should change. | the opening text should be “Kings Pawn Opening” | As expected. | Evidence 7. |
| 2.5 | The board should show legal moves if enabled in settings and should not if disabled. | From starting position, with the e2 pawn selected, the UI should be updated. | Should show e3, and e4 as legal moves. If disabled, should not show any legal moves. | As expected. | Evidence 8. |
| 2.6 | If new game is clicked, the board should show that. | No data. | The board should be restored to the starting position and the move list should be cleared. | As expected. | Evidence 10. |
| 2.7 | The game should choose a move that it determines to be the best. | A position shown in the evidence, the queen is hanging, it is objectively the best move to capture. | The engine should say that capturing the queen is the best move. | As expected. | Evidence 11. |

3.3.0 Data storage & manipulation:

| Test No. | What is being tested | Data | Expected Result | Actual Result | Evidence Reference |
| --- | --- | --- | --- | --- | --- |
| 3.1 | The user should be able to choose settings, and this should be saved as a configuration file. | Selecting options on the settings form. | When the user presses save, the changes should be made on the form, and the settings file should be updated. | As expected. | Evidence 12. |
| 3.2 | The user should be able to create an account and it should be saved as its own file. The account-table should also be updated. | Creating an account with a username and password. | The account file should appear in the directory of the accounts table. | As expected. | Evidence 13, Evidence 9. |
| 3.3 | The ECO code file should be opened and used by the opening text to update the opening name. | Playing an opening, the opening text box should update. 1. e4.. | The opening text should read: Kings pawn Opening. | As expected. | Evidence 7. |
| 3.4 | The user should be able to open a PGN file using the play button. | Any PGN file. | The data form the PGN should be read into the board, and the board and move list should be updated appropriately. | As expected. | Evidence 14. |

3.4.0 Linkage:

| Test No. | What is being tested | Data | Expected Result | Actual Result | Evidence Reference |
| --- | --- | --- | --- | --- | --- |
| 4.1 | The user should be able to view the signup form when clicking signup. | Clicking signup on the interface form. | The signup form should open as a dialog. | As expected. | Evidence 1. |
| 4.2 | The user should be able to view the login form when clicking login. | Clicking login on the interface form. | The login form should open as a dialog. | As expected. | Evidence 9. |
| 4.3 | The user should be able to view the settings form when clicking on settings. | Clicking settings on the interface form. | The settings form should open as a dialog. | As expected. | Evidence 12. |
| 4.4 | The user should be able to select a PGN file when clicking play. | Clicking on play on the interface form. | An open file dialog should be shown, allowing the user to open PGN files. | As expected | Evidence 14. |

3.5.0 Overall System:

Throughout the development of my solution I have used iterative testing in order to test the overall system. In order to verify that this testing was sufficient, I will play through a game from beginning to end, providing screenshots of the application after each move, and including the move that the engine suggested.





fs





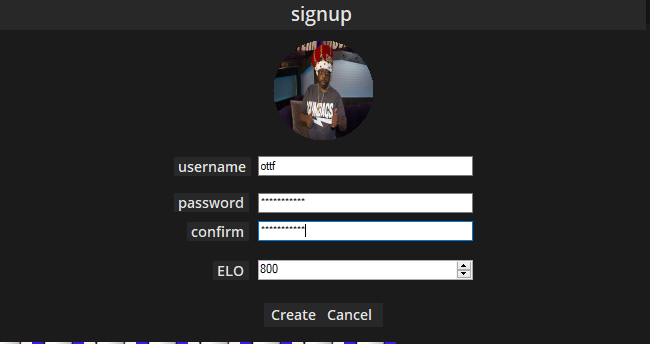
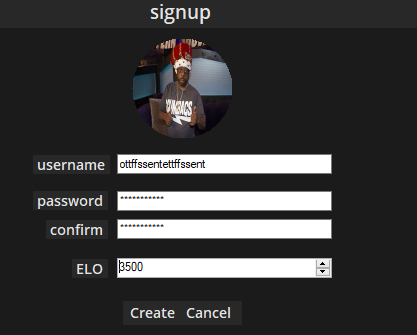
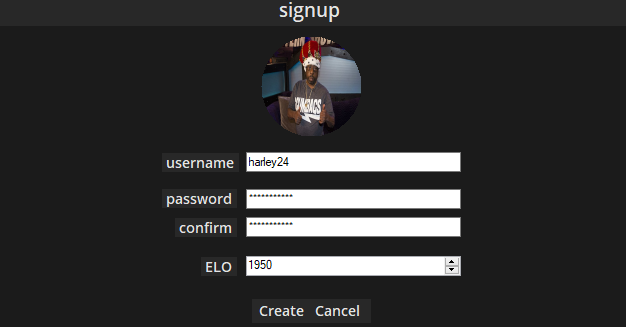
The sequence of moves shown in this test is the shortest game possible in a game of chess, and all moves shown are valid.

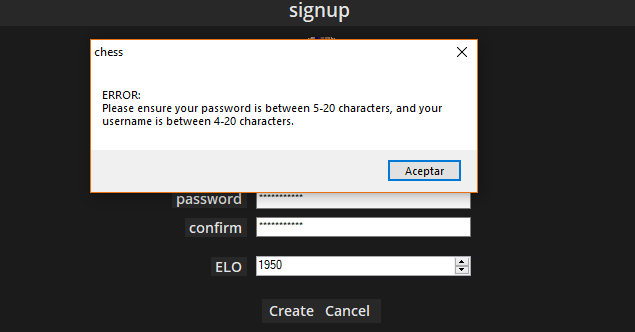
When the player clicks new game, the board is restarted, and they can play again as shown here.

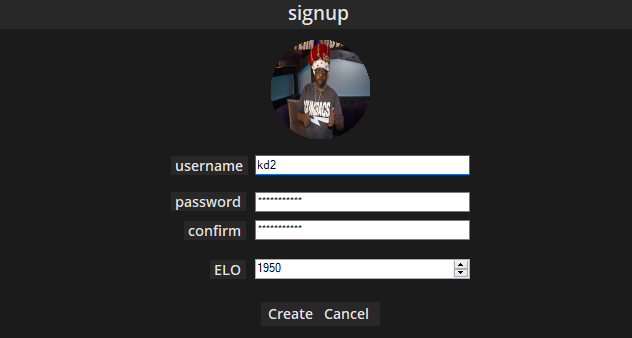
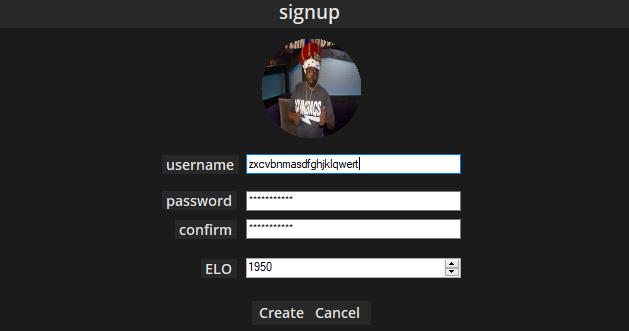


3.6.0 Proof:

Evidence 1:

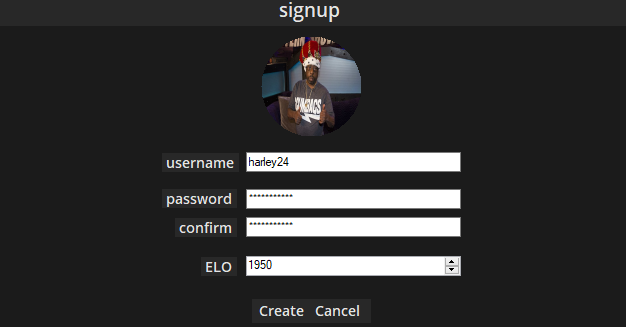
When the following data was input, the form closed, and the account was created. Pay attention to both ELO, and username.

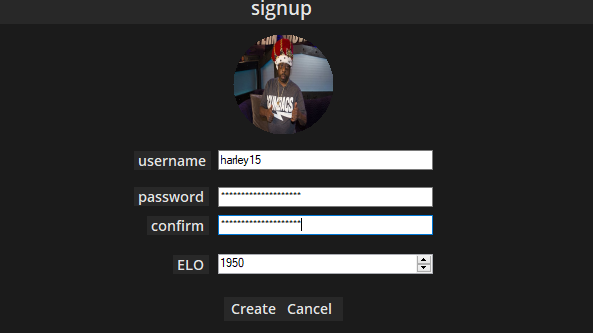
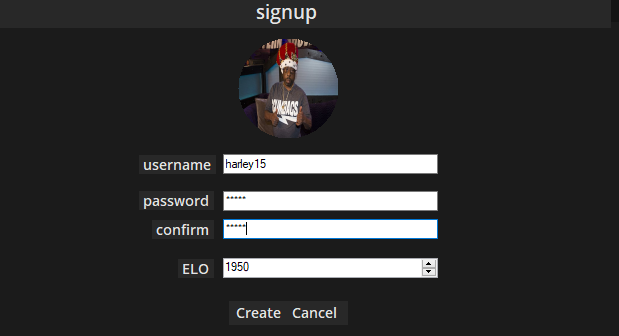
When the following data was input, the corresponding error message was shown.



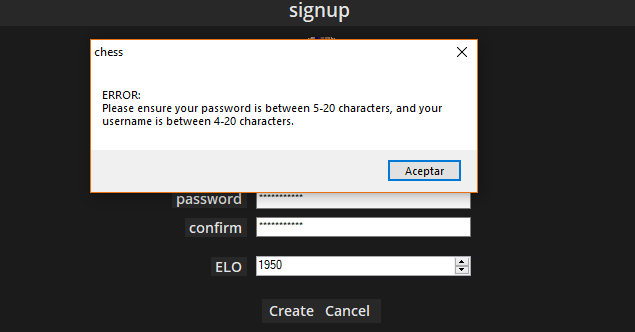
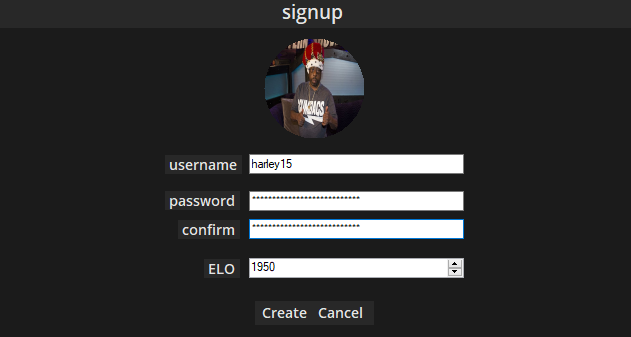
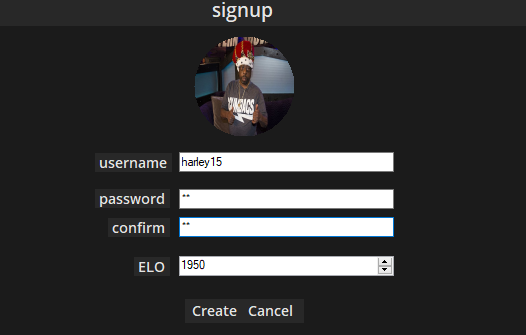
Evidence 2:

The following screenshots show evidence for extreme, extreme and valid data respectively. When create is clicked, the form is closed, and the account is created.



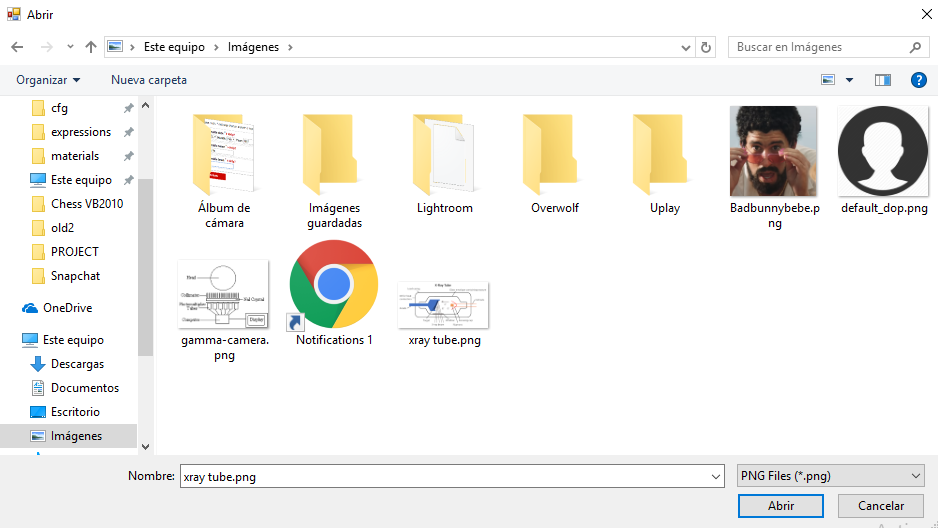


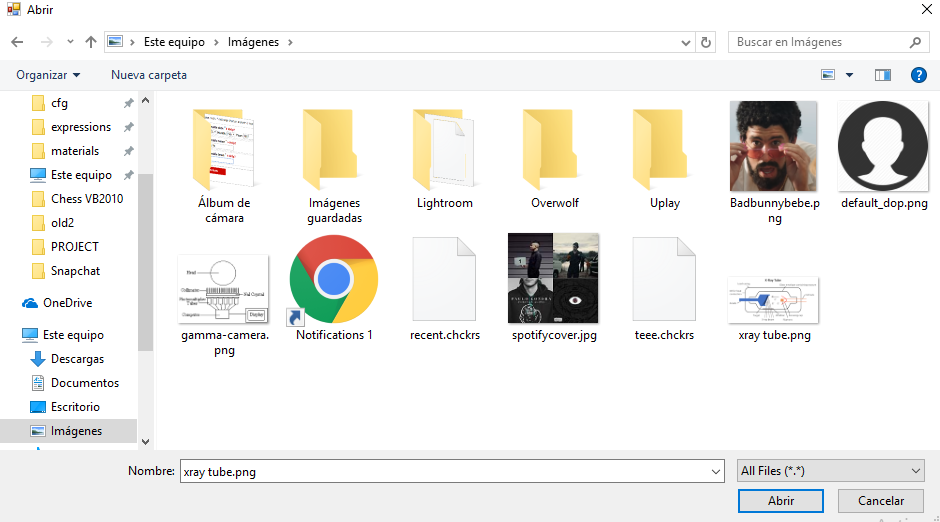
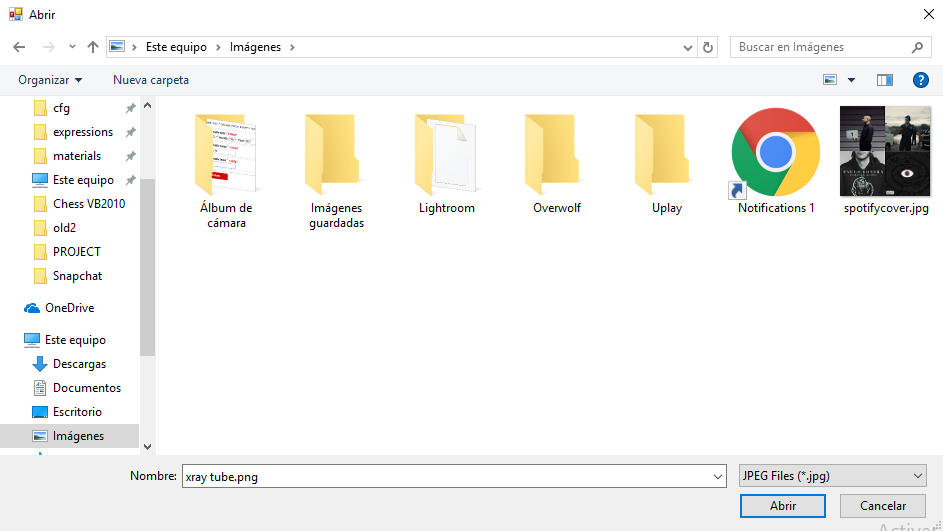
The following is invalid data, when create is clicked the following error message is shown.

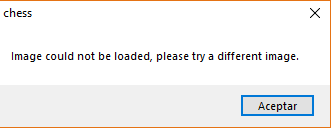


Evidence 3:

The following shows the file dialog when the image selection is clicked. As you can see, a JPG / PNG filter is used. If the all files filter is used, and a non-JPG/PNG file is selected the following error message is shown.

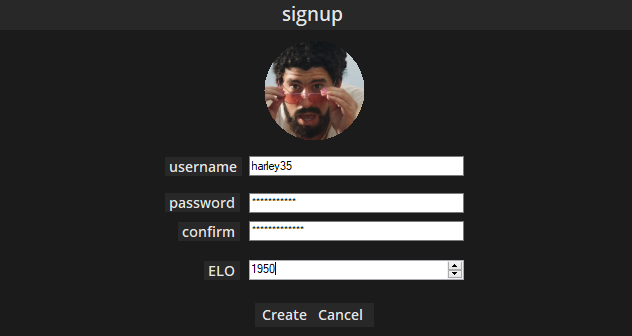


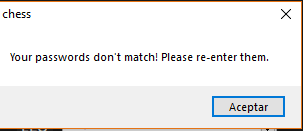


This occurs when teee.chckrs is selected.

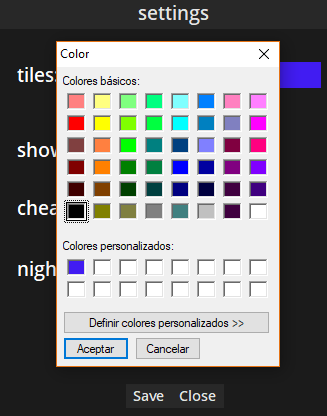
Evidence 4:

Here, the passwords entered in both text boxes are of different length, and so are different. The following error message is displayed.

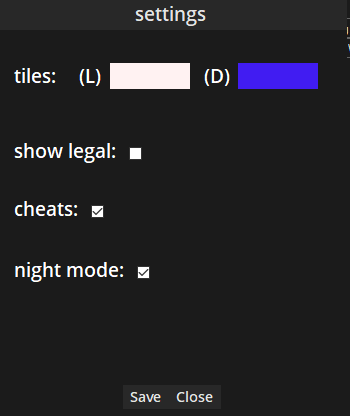




Evidence 5:

When the user tries to change a tile colour, they are given the option to do so through a choose colour dialog, therefore the only possible input is an RGB colour.

Evidence 6:



The user can only make an input using a checkbox, which is self-validating.

Evidence 7:

In the following tests, the user tried to play e4, and Qe4 respectively. The resulting board is shown. As evident, e4 is played and the board is updated, whereas Qe4 is not played because it is not a legal move in the position.

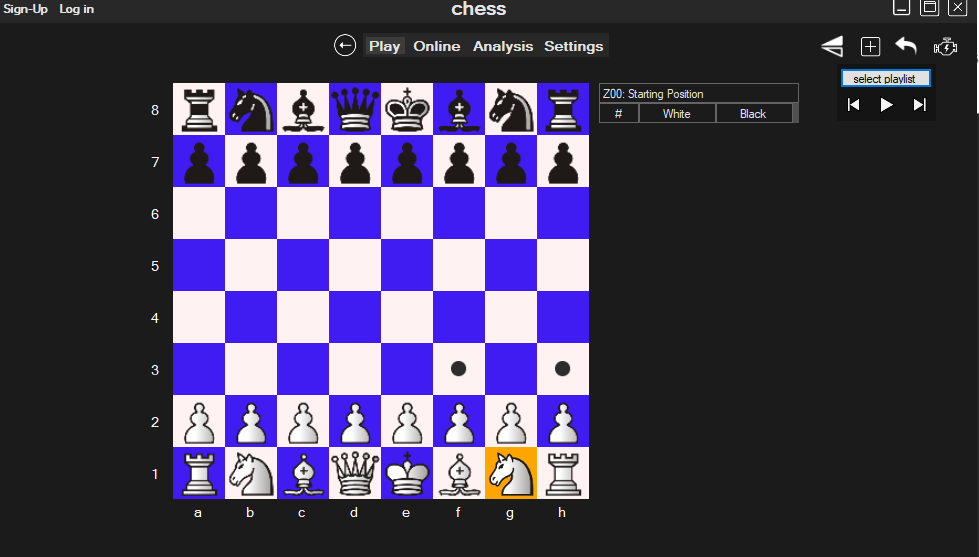




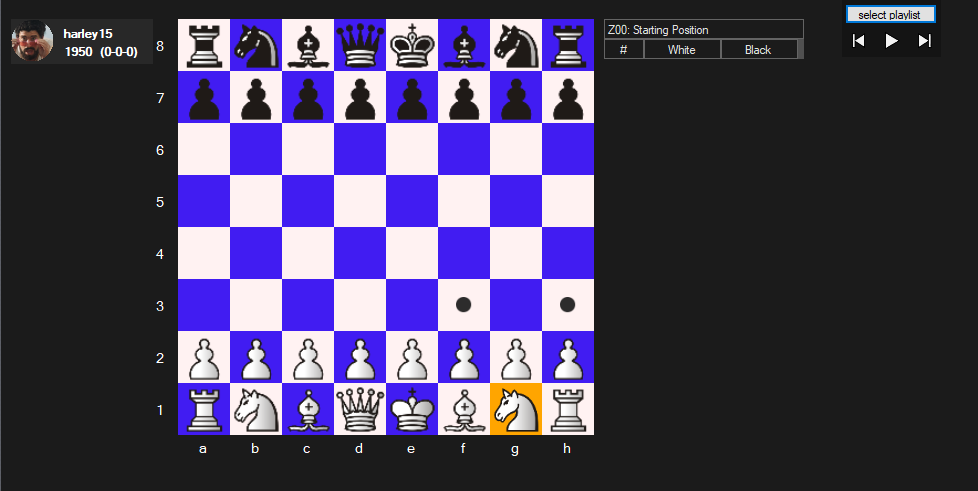
Evidence 8:

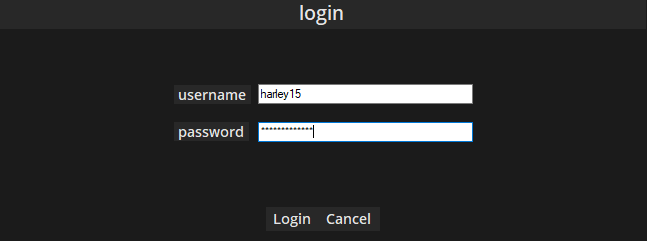
The following shows the legal moves generated by the game, and how they are shown on the board with show legal moves enabled.





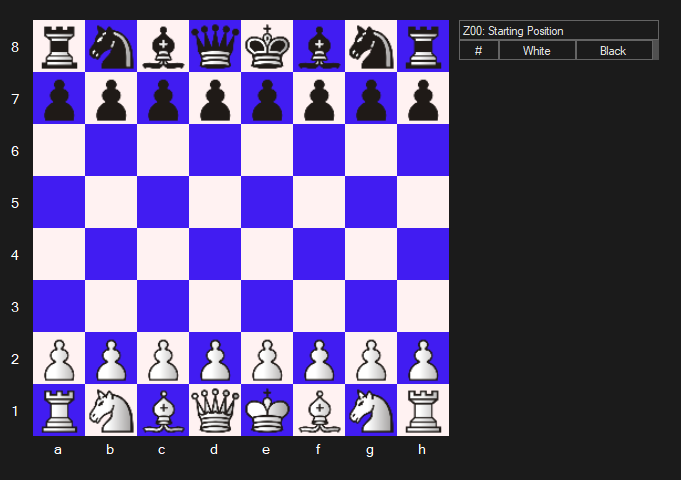
Evidence 9:

When correct details are entered, the window closes, and the account window is shown.



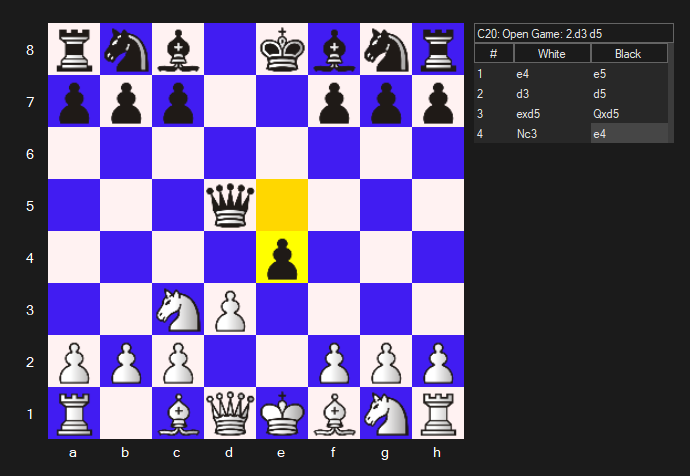
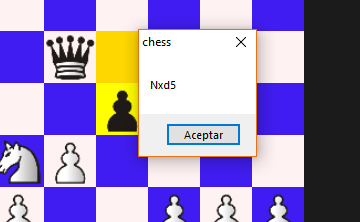
Evidence 10:

In the following position, the new game is clicked. The result of clicking this is shown next to the original position. 

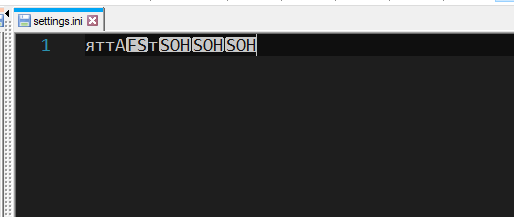


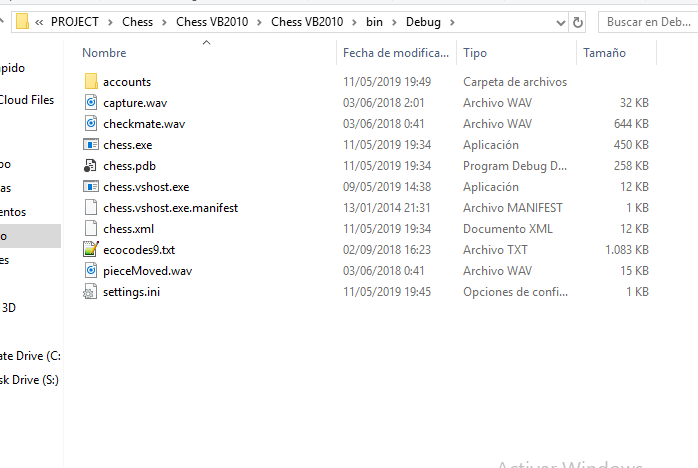
Evidence 11:

In the following position, the best move is to capture the queen as white. The engine provided the following move when the Minimax algorithm was executed.



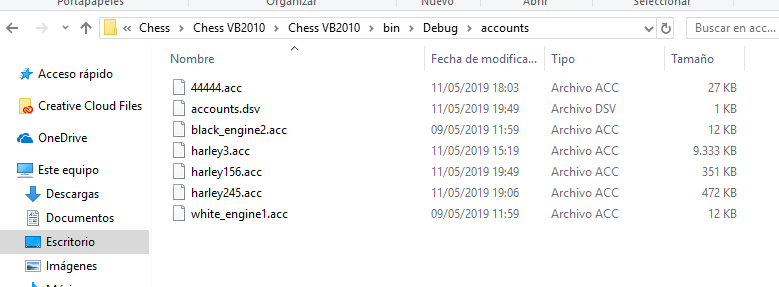
Evidence 12:

After selecting settings, the following settings file was created in the directory of the executable of the program. It is evident that the program saved properly, since when I closed and re-opened the application, the same settings were applied.

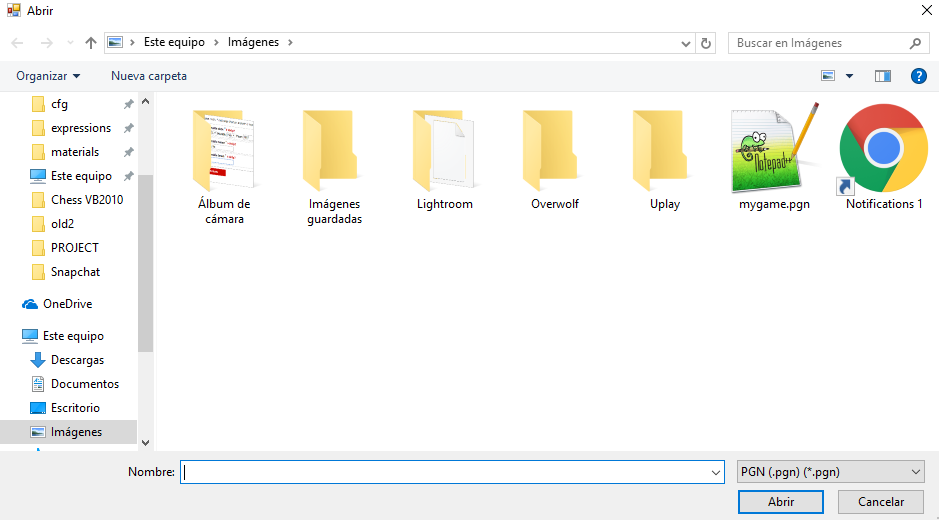


Evidence 13:

Here it is shown that an account file was made. Evidence 9 shows that the account file was able to be understood by my program when the user logged on.



Evidence 14:

When the play button is clicked the following open-file dialog is shown. As evident, a filter is used to only show .pgn files. The result of loading the file is shown in the second image, where the position from the PGN is loaded onto the board.

