# **Evaluation: Evaluative Testing:**

## Post-Development Testing

The majority of my evidence will be screenshots but some will be sections of video of the game working. Here is the link to this video:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Aspect Being Tested** | **Justification of test** | **Type of test** | **Test Data Used** | **Expected Result** |
| 1 | Different Difficulty Settings | Needed for stakeholder (casual chess player). Success criteria item 1 | Valid | Clicked radio buttons then looked for a change in response time for the computer move | The response time should change so that it is approximately the same as the listed time (e.g. medium is 5 seconds) plus an extra second delay. |
|  | **Actual Result** | **Evidence** | | | **Success?** |
|  | This did cause a notable change in response time for the computer move. | I can change the radio button easily as shown. The response time change was noticeable. **See video clip 1**. | | | Full Success |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Aspect Being Tested** | **Justification of test** | **Type of test** | **Test Data Used** | **Expected Result** |
| 2 | Client Side Validation | Test that I cannot input an invalid move or move the opponents pieces on my turn.: Success Criteria items: 2 and 28 | Invalid | Clicked on my piece than an invalid square, then tried to click and move the opponent’s pieces | The invalid click events are ignored, they will only affect the highlighting. |
|  | **Actual Result** | **Evidence** | | | **Success?** |
|  | The result was as expected, bogus click events were ignored | **See clip 2** | | | Full Success |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Aspect Being Tested** | **Justification of test** | **Type of test** | **Test Data Used** | **Expected Result** |
| 3 | Client Side Validation | To meet success criteria items 3 and 29, the user cannot move when it is not the user’s turn. The move that the computer produces should then be valid | Invalid | Set difficulty to high and then try to move while it is the opponents turn.  The computer then made a move which was valid | These clicks will be ignored and there will be no corresponding output.  The computer’s move should be a valid legal move |
|  | **Actual Result** | **Evidence** | | | **Success?** |
|  | The expected result, invalid inputs were ignored. The move that the computer made was valid | **See clip 3** | | | Full Success |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Aspect Being Tested** | **Justification of test** | **Type of test** | **Test Data Used** | **Expected Result** |
| 4 | Check: testing that a message is used to show check and testing that the user cannot input invalid moves that cause their king to be in check | To meet success criteria items 4 and 11 and to ensure that the program has the relevant and does not allow invalid inputs | Valid (for message)  And  Invalid (try to make invalid move | I play the game until I had moved my king into check | The computer should make use of the main title to show that it is my go and I am in check. It should show with highlighting that I cannot make moves that violate the rules of check. When I try to make these moves anyway, they should be ignored |
|  | **Actual Result** | **Evidence** | | | **Success?** |
|  | The title message identified check when if occurred.  I couldn’t move in a way that caused check.  When I was in check, the highlighting showed the correct legal moves, I couldn’t input an invalid legal move | **See clip 4** | | | Full Success |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Aspect Being Tested** | **Justification of test** | **Type of test** | **Test Data Used** | **Expected Result** |
| 5 | All aspects of highlighting in both the user and computer move.  As well as checking that the user and computer can move | This will indicate if my usability feature to show the moves and legal moves with highlighting is working. I will also see it the computer can make a move and if it shown with highlighting  This corresponds to items: 5,6,7,8 from the success criteria | Valid | I implemented user moves step by step to see the highlighting at each part of the process. | I can click one of my pieces, it will be highlighted red and every where it can move to will be highlighted green. The computer move will be highlighted before it happens. The moving piece will be red and the square it move to will be green. |
|  | **Actual Result** | **Evidence** | | | **Success?** |
|  | Exactly as expected | **See clip 5** | | | Full success |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test No.** | **Aspect Being Tested** | **Justification of test** | **Type of test** | | **Test Data Used** | **Expected Result** |
| 6 | The Chess AI should be capable of making smart moves, it should also be able to learn over time.  I will also try to show that I have improved the algorithms efficiency by using multiple codes | Tests success criteria item 9,26, 27 | Valid | I will make a series of moves to lay a trap where it trades a knight for a pawn | | on trivial difficulty it should fall for the trap. Then on extreme difficulty and all later games it should avoid the trap. I should be able to show this with the database. |
|  | **Actual Result** | **Evidence** | | | | **Success?** |
|  | As expected, at a greater difficulty it avoided the trap. This cache then allowed a low difficulty to also avoid the trap | **See clip 6** | | | | Full Success |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Aspect Being Tested** | **Justification of test** | **Type of test** | **Test Data Used** | **Expected Result** |
| 7 | Computer’s ability to put me in checkmate. And the appropriate game over output | This will test that the AI is trying to actually win and will test how the program uses outputs to clearly show when the game is over. This corresponds to success criteria items 10, 12, 13 | valid | I will try to put myself in checkmate | The computer should be able to put my king in checkmate. The main title should then show that I am in check mate. The board should then be disabled |
|  | **Actual Result** | **Evidence** | | | **Success?** |
|  | As expected, the AI took the option of putting me in checkmate over other options like trying to take more pieces. Then the title indicate the game was over and the board was disabled | **See clip 7** | | | Full Success |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Aspect Being Tested** | **Justification of test** | **Type of test** | **Test Data Used** | **Expected Result** |
| 8 | Bulk testing of many features. I will test:  The move history and pieces lost tables work; the reset and concede buttons work and that reloading the webpage restores the game. I will also check that the server connection is fast enough | This will test various features are working as intended. It will test that the program meets success criteria item:  15, 30, 31, 32 | Valid | Checking the contents of the tables and clicking the buttons | The tables should be able to accurately keep track of the chess game (moves and pieces taken) as I make moves. The buttons should be able to end the game and then reset it. This includes resetting the move and pieces tables |
|  | **Actual Result** | **Evidence** | | | **Success?** |
|  | As expected the tables and buttons all worked | See clip 8 | | | Full success |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Aspect Being Tested** | **Justification of test** | **Type of test** | **Test Data Used** | **Expected Result** |
| 9 | Bulk testing of many features. I will test:  that reloading the webpage restores the game. I will also check that the server connection is fast enough. I will also verify that the program works in different aspect ratios. | This will test various features are working as intended. It will test that the program meets success criteria item:  17, 19, 25 | Valid | I started a new game, reloaded the tab. Then show the response speed and then different aspect ratios | I expect the response speed to be fast. The game should be reloaded when the tab is reopened.  The website should work on:   * PC / Laptop * Tablet Landscape * Phone portrait |
|  | **Actual Result** | **Evidence** | | | **Success?** |
|  | The reloading of the game worked fully. The latency from the server was low (response times were fast).  Some aspect ratios worked perfectly while others struggled to fully render the move history table but could render everything else. | **Clip 9** | | | Partial success  Some aspect ratios for mobile devices struggled to render the move history table. It was either too large of not fully shown. Other aspect ratios worked perfectly well. Test objectives 17 and 19 were fully achieved |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Aspect Being Tested** | **Justification of test** | **Type of test** | **Test Data Used** | **Expected Result** |
|  |  |  |  |  |  |
|  | **Actual Result** | **Evidence** | | | **Success?** |
|  |  |  | | |  |

## Usability Survey:

I had an extensive discussion with a stake holder that plays casual chess on a phone regularly. I asked him to play around with the final product and comment on how well it met his needs:

He said:

“The website has an overall clear appearance and it is clear what the pieces are.

Doing what I would do intuitively, I will click a pawn. It now shows where the pawn can move and after clicking it, it has now moved.

The highlighting makes it clear which black piece has moved and how it has moved.

I now realise that scrolling down there is extra information

It is clear how to change difficulty

The pieces taken and move history are clear. I understand exactly what they show.

I have made another move as expected.

I will now try to take a piece. I like how the piece is highlighted green.

I like the choice of cream and brown colours for the board, it makes the pieces distinct.

I will try conceding.

The game has stopped .

I will try to restart.

It is clear what the buttons are.

I will try closing the tab, the game is still there. That is really convenient neat as you can just continue the game

A possible improvement could be to make use of the space either side of the chess board in landscape

“

I found this feedback very useful. It acts as good evidence to suggest that I have succeeded in making the website clear and accessible (success criteria item 24).

I will make sure to use a few of the points that came up in the questionnaire:

* I will ask if it is intuitive: what to do when the website loads.
* I will ask if people immediately noticed the difficulty setting and tables below the buttons.

I used a google form to conduct a questionnaire of stakeholder to get their feedback on the usability features of the chess program:

Chart, bubble chart

Description automatically generated

This shows that the webpage is reliable.

Chart, bubble chart

Description automatically generated

This is expected as I put these elements in immediate view as they were the most important.

Chart, pie chart

Description automatically generated

This feedback show that the program would benefit from some kind of text or arrow to show that the use could scroll down for more information and settings.

Chart, pie chart

Description automatically generated

This likely show that the stakeholders I was able to collect data from are not a representative sample of the whole population as they are mostly using laptops to play with the website. A larger sample may show that more users will use tablets.

Chart, pie chart

Description automatically generated

This question shows that the input method for making a move was intuitive. This is likely partly due to the interface is similar to many other chess websites and programs that these stakeholders may have already encountered. It is encouraging as it shows that the program is easy to use.

**Chart, bubble chart

Description automatically generated**

This response shows that the cream and brown colours used for the board, combined with the text shadow were able to contrast the pieces against the board. This made them very clear to users. Chart, bar chart

Description automatically generated

This shows that having a range of different difficulty settings is a feature that is appreciated by casual chess players (the stakeholder / target niche). This allows people to play a easy of challenging game of chess, whatever their level. It is likely that the main issue that some people had with the difficulty setting was that each additional difficulty took longer to decide the computer’s move.

Chart, bar chart

Description automatically generated

This shows that the ability to reset the games was also appreciated. It seems that some people didn’t see a use for the concede game button.

Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

These results suggest that people had some issues with these tables. It is likely that the fact that they had to know to scroll down contributed to this. In addition the lack of chess squares ladled on the board (e.g. so that the user could easily find the square B5) means that some player cannot make full use of the information in the move history table.

Chart, bubble chart

Description automatically generated

This is very encouraging. It shows that efforts to maximise contrast and make font sized big enough have allowed users to clearly see what is going on.

Chart

Description automatically generated with medium confidenceThis shows that the highlighting is be far the most popular feature. Users seemed to like how it made moving pieces intuitive. It also helps new players who are not yet aware of where all the pieces can move. The use of highlighting to show the computer move seemed to also be appreciated as it drew the user’s eye and made the computer’s move clearer.

Chart, bubble chart

Description automatically generated

This shows that my testing and validation has paid off as the program is now very robust. None of the users had issues where the program crashed. This is inline with the results of my post development testing.

Chart, pie chart

Description automatically generated

As many of the stakeholders were only able to test the program for a limited time, some were not able to fully finish a chess game. However the 4 who were able indicated that they felt that the resulting message and the disables board was clear and intuitive.

Chart

Description automatically generated with medium confidence

This feedback is very encouraging. It show that I was able to meet my stakeholders main need: entertainment. All of my stakeholders enjoyed that game. This indicates that the game meets the needs of casual chess players well.

### Breakdown of success criteria (what has been achieved):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Number** | **Feature name** | **Importance** | **Achieved** | **Evidence** | **Evaluation** |
| 1 | Difficulty settings | essential | Fully | Post development test 1 (and also test 6) | I used radio buttons to input the difficulty settings. I created 7 different difficulties that affected how long the AI had to explore the decision tree. I felt this was important as casual chess players could vary in skill and may not always want a challenge. |
| 2 | Validation client side | Essential | Fully | Post development test 2 (and also test 3) | As soon as the website opens, the computer waits for the user’s move. The user is always white and has unlimited time to think about their moves. |
| 3 | Validation client side | Essential | Fully | Post development test 3 (and also test 2) | I used both client and server side validation to ensure only the right player can make a move at any given point in the game |
| 4 | Validation client side | Essential | Fully | Post development test 4 | I showed how the highlights showed that the user couldn’t make some moves. I then showed that any inputs what tried to make these invalid moves were ignored |
| 5 | Intuitive Move Input with dot markings | High priority | Fully | Post development test 5 | I implemented this feature fully and it was well received by users. It makes it clear where the user can and can’t move a piece. It also helps users who are not that familiar with the game of chess. |
| 6 | Graphically show the move | Essential | Fully | Post development test 5 | Once the move is inputted, the new board state is shown to the user (with their piece moved to the new square). There is a purposeful delay before the computer move is added to ensure that I don’t have too much happening at once. This lets me clearly show the user that their move has been implemented |
| 7 | Computer Move output | Essential | Fully | Post development test 5 | The computer is able to move the black pieces on the board to successfully act as an opponent to the user. |
| 8 | Highlighting | High priority | Fully | Post development test 5 | Highlighting for a short time shows what piece the computer is moving and where the piece is going. Stakeholders found this feature helped them see exactly what move the computer had made. |
| 9 | Chess engine mid game | Essential | Fully | Post development test 6 | The Chess AI was able to make “smart” decisions when it was able to sufficiently search the tree. This allowed it to avoid losing its pieces and to take my valuable pieces |
| 10 | Chess engine late game | Essential | Fully | Post development test 7 | The Chess AI is able to pursue checkmate in the late game, once it already had a significant piece advantage. This help the user experience as there AI opponent is really trying to win. This should make the game more inresting. |
| 11 | Alert after each move for check | Essential | Fully | Post development test 4 | I made use of a large title to show which users turn it was as well as to make it clear, when a player was in check. I believe this provided some necessary direction to users (about whose turn is was). |
| 12 | Proper recognition of game over and its properties | Essential | Fully | Post development test 7 | I again used the main title element to output to the users that the game was over. I also disabled the board. This allowed users to still see all the pieces, without being able to move them. |
| 13 | Chess engine aware of different game over outcomes | Essential | Fully | Post development test 7 | It is able so successfully decide to pursue winning by checkmate over taking valuable pieces. It will also avoid losing by checkmate at all cost. It is aware of stalemate (normal and 3 repeat stalemates). It will try to avoid a stalemate when it is winning and cause one when it is losing |
| 14 | Undo button for last user move | Low priority | Not Implemented |  | This would have been a nice feature as it could have helped users who miss-clicked. In the end I already planned to implement so many other features which meant this feature would only be implemented if I had time to spare. |
| 15 | Log of history of moves in game | Ideal / desirable | Fully | Post development test 8 | This feature works fully. It logs the user’s and computer’s moves in a table. This includes whenever a piece is taken. The feature could be improved by adding numbers and letters down the side of the chess board to allow new players to locate squares that it refers to. |
| 16 | Step back through log | Low priority |  |  | This functionality would have required significant time to implement as additional complexity would have to be added to both the front end and the backend. |
| 17 | Fast and reliable connection between front and backend | Essential | Fully | Post development test 9 | The server and client browser are connected by WebSocket connections which hare extremely fast due to using a direct TCP/IP connection. This meant that the final product didn’t feature any lagging. |
| 18 | Allowing users to login & sign up | Low priority | Not implemented |  | This feature would have required huge amount of additional work. I would have needed to update the database; front end and backend validation; made a new python module to handle users and make significant changes to the user interface to allow for a login system. As I was able to implement a save game feature without users, it hardly seemed worth the steep time and opportunity code |
| 19 | Reloading unfinished game | Ideal / Desirable | Fully | Post development test 9 | This functionality was added using cookies and a database. I am proud of this feature as I think that it is technically impressive and very useful. |
| 20 | Puzzles | Low priority | Not implemented |  | This feature was written of early in the design process for being too much work. |
| 21 | Draughts | Last add, very low priority | Not implemented |  | This feature was written of early in the design process for being too much work and deviating from chess |
| 22 | Allow users to start with a custom game layout | Low priority | Not implemented |  | This would have been a nice feature to add. I could have allowed the user to be black and go second. I think that overall if makes sense for the user to go first as the computer has many advantages (such as looking at move possible board states). As such, I don’t think that much was lost in not implementing this feature |
| 23 | Leader board and trophy system | **After interview I no longer want to implement this (lowest possible priority)** | **Not implemented** |  | I wrote this feature off early in the analysis section. I believe that the idea of a leader board is more suited to competitive players than casual players. |
| 24 | Accessibility through website | Essential | Fully | I cite the usability interview and questionnaire as evidence as they show that the website is clear and intuitive | I aimed to ensure that the layout of the page and color, size and font of the text were as clear as possible. I prioritised this over aesthetics (hence the background color is white). I think that maximising clarity will benefit all stakeholders and the programs usability as a whole. I particularly think that it will benefit older and less tech-savvy stakeholders. |
| 25 | Mobile access | Desirable | Partially | Post development test 9 | I think that the efforts I did make to achieve this feature were still important. The final product should work well enough on most aspect ratios. This should allow the elderly who struggle to use PCs to play on tablets. This should also allow people on the go to play on their phones. |
| 26 | Efficient adversarial AI | Highly desirable | Fully | See detailed explanation of improvements in prototype 3 (such as parallelisation).  The use of parallelisation and database cache was shown in post development test 6 | I am proud of my effort here. While I still am not able to quickly perform a depth 3 search, I brought the time needed to perform a depth 2 search of the stating positions from around 25 second to 12 seconds. I think that making depth 2 more possible is important as this lets the computer apply game theory to anticipate the user and make interesting decisions (like when we saw it avoid the trap). |
| 27 | The AI adversary feels as organic and real as possible | Highly desirable | Fully | Post development test 6 | I think that allow with making depth 2 searches more possible, and use of a database allowed allows the chess AI to learn make the AI a move real and dynamic opponent that is more fun to play against. |
| 28 | Validation of users move | Essential | Fully | Post development test 2 | I added both client and server side validation to ensure that the user couldn’t make an invalid move. |
| 29 | Valid computer move | Essential | Fully | Post development test 3 | I added checks within my code to catch erroneous and invalid moves made by the minimax algorithm and prevent them reaching the user and causing unexpected behaviour. |
| 30 | Pieces Taken output | Desirable | Fully | Post development test 8 | I added this feature as a way for users to see at a glance who is winning. This can be helpful information that a casual player might want. |
| 31 | Restart game button | Essential | Fully | Post development test 8 | The restart button allows the user to play a fresh new chess game. This can be done when the game is ongoing or when it is over. This allows users to start again if they want to (for example if they lose their queen). |
| 32 | Concede game button | Essential | Fully | Post development test 8 | The idea of this button was that it could stop the game progressing while still allowing the user to see all the pieces and the move history. This would stop the computer’s move coming through and changing the board state. Some players may also want the ability to concede (as they can in a real chess game). |

I have implemented all the high priority features in the success criteria. The low priority features were not completed due to time constraints.

## Limitations and Maintenance:

This program is only a prototype and has various limitations that could get in the way of it being fully deployed. These include limitations of the program itself and difficulties in maintaining the system at scale.

Limitations with the program itself include:

* One of the main limitations of the program is that the database is currently all stored withing one file. This would pose scalability problems if the program were to be used by many people. The solution would be to use a more scalable database system. For instance, the database could be implemented using a cloud based system such as AWS. This would make the database far more scalable and able to hold many more unfinished games
* Another limitation is that I didn’t implement all the move of chess. I did implement pawn promotion but didn’t implement moves like en passant and castling. If this chess program were to become as popular as come of its competitors like Lichess, it would need to have these moves implemented.

As part of maintenance, certain issues may arise:

* For one, the database may accumulate to many unfinished games. While each one is only around 4KB of binary data, this could slow that database and present an unnecessary cost if it got out of hand in future. To fix this, binary data representing chess games could be stored in files instead. The file names could then be stored in the database. In addition, a new column could be added called expiry date which ensures that saved games are not stored forever.
* Another issue is that the database may gather so many cached results that is has a very strong opening at all difficulties. This could erode the concept of difficulty settings and prove a problem for new players.

Some of the key ways in which this chess program could be improved:

* Additional features from my success criteria that were not implemented could be added to the program later.
* This could include implementing challenges and puzzles (e.g. games that are 3 moves away from checkmate)
* User login could also be added to allow users to progress through challenges, access saved games on multiple devices or share their progress