./

Learning Report – SDLC (CHESS)

Course Code: <CODE>



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
| 1. | 19/09/2020 | NAMAN CHOUDHARY |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

# 

Contents

[Checklist 3](#_Toc51401461)

[Activity and Tasks 4](#_Toc51401462)

[Activity 1– System/Software Development 4](#_Toc51401463)

[Activity 2 –CI Workflow for C Programming 4](#_Toc51401464)

[Activity 3 – Agile Aspects 4](#_Toc51401465)

CHESS ONLINE APPLICATION 5-8

Requirements 5

Test Plan 6

UML Diagram 7

GITHUB 8-9

References 10

# Checklist

* Installation of SW on Phone and Desktop
* Additional Aspects …

# Activity and Tasks

## **Activity 1**– System/Software Development

* Sub Tasks
* Complete and Evolve

## **Activity 2** –CI Workflow for C Programming

* Sub Tasks
* Complete and Evolve

## **Activity 3** – Agile Aspects

* …..

**ONLINE CHESS APPLICTION**

**Requirements:**

1. Users shall be able to connect via IP address.
2. Users shall be able to start a game once two users are connected.
3. Users shall be given the choice of who plays black and white.
4. Each user is to have their pieces start at the bottom of the board on their display.
5. The player playing white is first to move.
6. A player may forfeit at any time during gameplay.
7. A player must be given a confirm dialog before forfeiting.
8. Forfeiting shall end the game immediately.
9. The active player shall select a piece by clicking it.
10. When a piece is selected, all legal moves for that piece are highlighted.
11. When a piece is selected, the active player may select another piece by clicking it.
12. A selected piece must always belong to the active player.
13. The active player shall move the selected piece by clicking on any legal square.
14. The active player shall capture a piece by moving onto a legal square containing an opposing piece.
15. Captured pieces shall be displayed in a captured pieces box.
16. The inactive player may request to undo the prior move.
17. There shall be no more than one undo request per turn.
18. An undo request shall be ratified by the active player.
19. When an undo request is accepted the game is reverted to the state of the board prior to the request.

20. A player shall be able to save a log of the moves at any time. Priority 2

**TEST PLAN (AGILE)**

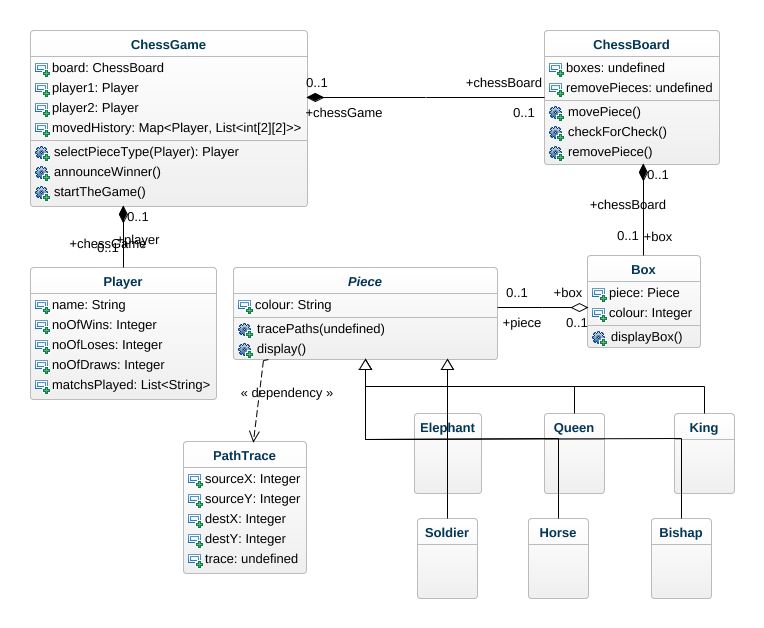
**LOW LEVEL:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SL  No | DESCRIPTION | PRE-CONDITIONS | EXPECTED  INPUT | EXPECTED  OUTPUT | ACTUAL  INPUT |
| 1. | Check that program compiles error free | Net connectivity is intact | The application should respond for every move | The prog compiled without errors | The prog compiled without errors |
| 2. | Execute the applet | Give commands for the applet | Give commands for the applet | Applet displayed | Applet displayed |
| 3. | Check proper ordering | All the pieces must be in order | Pieces put in place | Performed | Performed |
| 4. | Check the movement of pawn | Pawn must initially in position | Pawn moves two steps if in initial position or move one step | Moved two steps ahead | Moved two steps ahead |
| 5. | Check the movement of knight | knight must initially in position | Knight move 21/2 steps ahead or back | Moved 1 steps ahead | Illegal move |
| 6. | Check the movement of rook | Rook moves either vertical or horizontal | Rook is moved five steps vertical | Moved five steps vertical | Moved five steps vertical |

**HIGH LEVEL:**

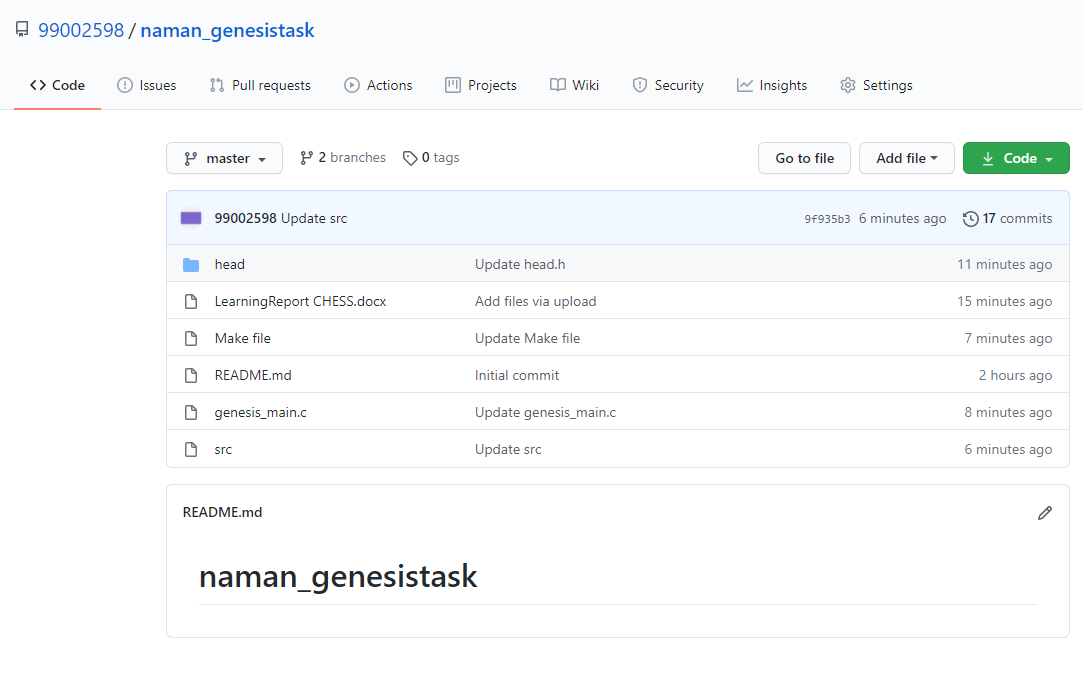
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SL  No | DESCRIPTION | PRE-CONDITIONS | EXPECTED  INPUT | EXPECTED  OUTPUT | ACTUAL  INPUT |
| 1. | Check the  movement | Game already started | Move of rook by the black | White knight killed | White knight killed |
| 2. | Check the total Kills | Half-way through the game | White take a move | White 10 kills  Black 8 kills | White 10 kills  Black 8 kills |
| 3. | Check for winner | Checkmate | Black takes a move | Black wins | Black wins |

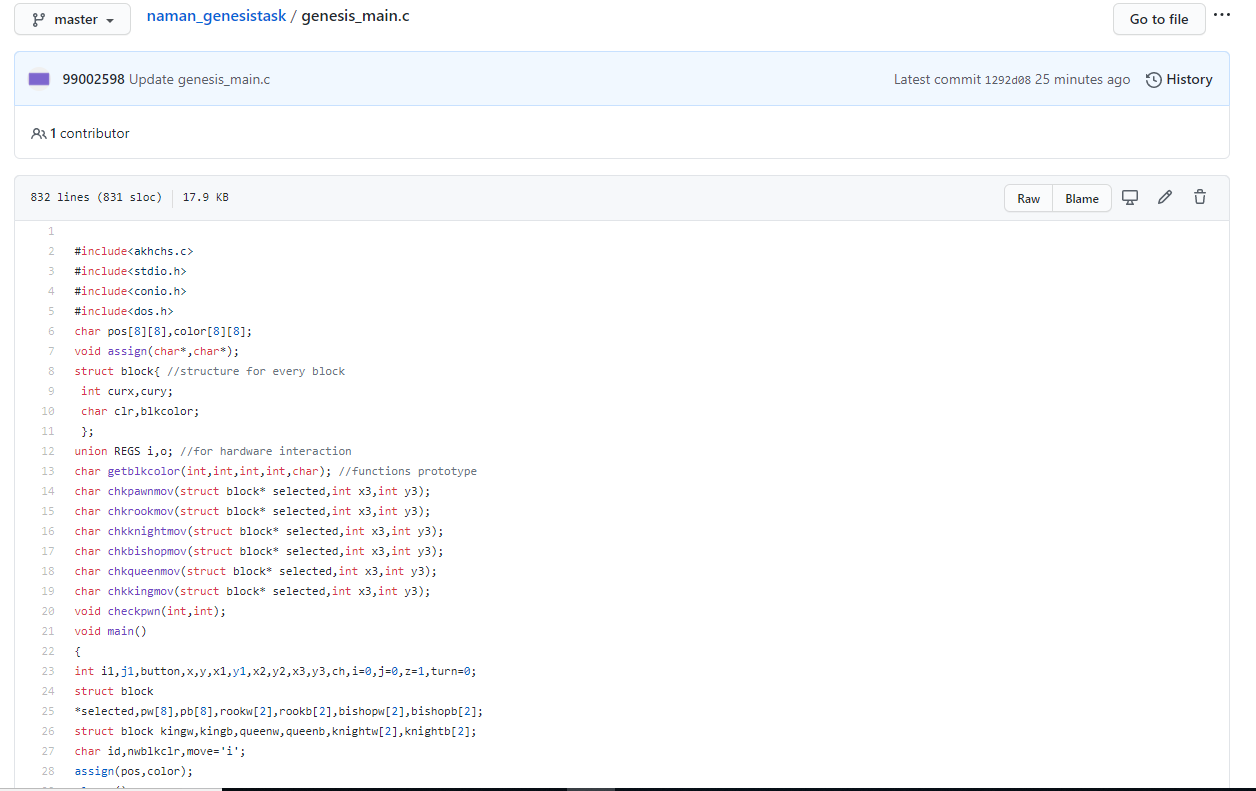
**UML DIAGRAM**



**GITHUB**

**(chess Application)**





**CHESS APPLICATION- 850 lines multifile code**

**REFERENCES**

1. Wikipedia: Chess template/diagrams

<https://en.wikipedia.org/wiki/Template:Chess_diagram/testcases>

1. Reddit: chess simulator test case

<https://www.reddit.com/r/chess/comments/eztufo/chess_simulator_test_cases/>

1. Students project for chess game board

<http://www.studentsproject.in/2015/06/test-case-for-chess-game-document-sample-be-projects.html>

1. Creativity chess diagrams

<https://creately.com/diagram/example/im3l3lv02/Chess>

1. Software requirement specifications for chess By Mayra Aguas, Alfred Blackman, George D'Andrea, Paul Vu, and Gurvinder Singh.

<http://www.pages.drexel.edu/~pv42/thebiz/SRS-1.pdf>

1. Creativity chess diagrams

<https://creately.com/diagram/example/im3l3lv02/Chess>

1. GITHUB Repository

<https://github.com/99002598/naman_genesistask.git>