MVJ COLLEGE OF ENGINEERING
Since 1982

An Autonomous Institute
(Affiliated to Visvesvaraya Technological University, Belagavi
Approved By AICTE, New Delhi,
Recognized by UGC under 2(f) & 12(B)
Accredited by NBA and NAAC)

A MINI-PROJECT REPORT
ON
ONLINE CHESS GAME

Submitted in partial fulfilment of requirements for the award of 6th Sem degree,

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Submitted By:
AMULYA
1MJ20CS021

Under the Guidance of
Mr Vinay Raj A S
Assistant Professor, Department of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MVJ COLLEGE OF ENGINEERING
BANGALORE-67
ACADEMIC YEAR 2023-24

MVJ COLLEGE OF ENGINEERING

Near ITPB, Whitefield, Bangalore-67

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MVJ COLLEGE OF ENGINEERING
Since 1982
An Autonomous Institute
(Affiliated to Visvesvaraya Technological University, Belagavi
Approved By AICTE, New Delhi,
Recognized by UGC under 2(f) & 12(B)
Accredited by NBA and NAAC)

CERTIFICATE

This is to certify that the mini-project work, entitled "**ONLINE CHESS GAME"** is a bonafide work carried out by AMULYA(1MJ20CS021) in partial fulfilment for the award of the degree of Bachelor of Engineering in Computer Science & Engineering during the academic year 2022-23. It is certified that all the corrections/suggestions indicated for internal Assessment have been incorporated into the Report. The mini-project report has been approved as it satisfies the academic requirements.

Signature of Guide
Mr. Vinay Raj A S

Signature of the HOD
Dr Kiran Babu T.S

Name of examiners:
1.
2.

Signature with date:

MVJ COLLEGE OF ENGINEERING

Whitefield, Near ITPB, Bangalore-67

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MVJ COLLEGE OF ENGINEERING
Since 1982
An Autonomous Institute
(Affiliated to Visvesvaraya Technological University, Belagavi
Approved By AICTE, New Delhi,
Recognized by UGC under 2(f) & 12(B)
Accredited by NBA and NAAC)

DECLARATION

I, **Amulya** hereby declare that the entire work titled "**ONLINE CHESS GAME"** embodied in this mini project report has been carried out by me during the 6th semester of BE degree at MVJCE, Bangalore under the esteemed guidance of **Mr. Vinay Raj A S**, Assistant Prof, Dept. of CSE, MVJCE. The work embodied in this dissertation work is original and it has not been submitted in part of full for any other degree in any University.

AMULYA ____________________
1MJ20CS021

Place:

Date:

ABSTRACT

The Online Chess Game project is a web-based platform that allows chess enthusiasts to play the game against each other over the internet. The project's primary objective is to provide a user-friendly and seamless experience for players to connect with each other from different parts of the world.

The platform's core feature is a virtual chessboard that allows players to move pieces by dragging and dropping them on the board. The software validates each move, ensuring fair play and adherence to the rules of the game. The game also includes features like in-game chat rooms, player profiles, and leaderboards to enhance the overall gaming experience.

The project's development involves various stages, including designing the user interface, developing the game's core functionality, integrating various features like multiplayer mode, leaderboards, and chat rooms, and testing the platform for bugs and errors.

The project aims to attract chess enthusiasts from different parts of the world and build a thriving community of players. The platform's success will be measured by the number of active users, positive feedback from players, and participation in organized tournaments and events. Overall, the Online Chess Game project aims to provide an accessible and enjoyable chess-playing experience to players of all skill levels.

ACKNOWLEDGEMENT

With gratitude, I acknowledge all those whose guidance and encouragement served as a beacon of light and crowned our effort with success.

I am thankful to the **Management of MVJ College of Engineering Bangalore** for their continuous support and encouragement in carrying out the mini-project work

I am thankful to our **Dr. M. Brindha,** Principal, MVJCE, Bengaluru for being a constant inspiration and providing all the facilities that were needed throughout the mini-project work.

I also like to express our sincere gratitude to **Dr M A Lourdu Antony Raj**, Registrarand Controller of Examinations, MVJCE, Bengaluru, for persistent guidance.

I am thankful to our **Dr Kiran Babu T.S**, HOD of Computer Science Engineering, MVJCE, Bengaluru, for being a constant support and providing all the facilities that were needed throughout the mini-project work.

I consider it a privilege and honour to express our sincere gratitude to our guide **Mr Vinay Raj A S, Assistant Professor,** Dept. of CSE, MVJCE, for his encouragement which has been a constant source of motivation to us for the successful completion of our project.

It’s also a great pleasure to express our deepest gratitude to all the other faculty members of our department for their cooperation and constructive criticism offered, which helped us a lot during our mini-project work.

Finally, I would like to thank all our family members and friends whose encouragement and support was invaluable.

Thanking you

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Online chess games have brought the ancient game of chess into the digital age, providing a platform for players to compete, learn, and engage with fellow chess enthusiasts from around the world. Chess, known for its strategic depth and intellectual challenges, has found a new home in the online realm, offering convenience, accessibility, and a vibrant community.

Online chess games provide a virtual chessboard where players can make their moves, analyze positions, and test their skills against opponents of varying levels. Through intuitive user interfaces and interactive features, players can engage in thrilling matches, participate in tournaments, and connect with a global network of chess players.

The virtual environment of online chess games opens up a realm of possibilities, allowing players to play at their own pace, choose opponents of similar skill levels, and access a wealth of learning resources. Online chess platforms often offer tutorials, puzzles, and lessons for players to improve their understanding of the game's mechanics, strategies, and tactics.

Beyond the gameplay experience, online chess games foster a sense of community. Players can join chess clubs, participate in forums, and engage in friendly banter with fellow enthusiasts. The online chess community provides a space for sharing knowledge, discussing strategies, and connecting with players who share a passion for the game.

Moreover, online chess games have become an avenue for professional players to compete and showcase their skills. With the rise of live broadcasting and streaming, viewers worldwide can follow high-profile chess

Overall, online chess games have transformed the way people play and experience chess. They have made the game more accessible, provided learning opportunities, and created a global network of chess enthusiasts. Whether playing for fun, learning, or competitive endeavours, online chess games offer a dynamic and engaging platform for players to immerse themselves in the fascinating world of chess. tournaments and matches, adding to the excitement and popularity of the game.

CHAPTER 2

PROBLEM ANALYSIS

- In online chess games, several problem areas can be identified that affect the overall user experience and gameplay. Here are some key problem areas in online chess games:
- Network Latency and Connectivity: Online chess games rely on stable internet connections, and issues such as network latency, connection drops, or slow response times can disrupt gameplay. Delays in move execution can lead to frustration and an unfair advantage for one player.
- Cheating and Fair Play: Online chess games face challenges related to cheating and fair play. Players may use computer assistance or other unethical means to gain an advantage, compromising the integrity of the game. Implementing effective anti-cheating measures and fair play policies is crucial to maintain a healthy and competitive environment.
- User Interface and User Experience: The user interface and user experience of online chess games can impact player engagement. Poorly designed interfaces, complex navigation, or unintuitive controls can make it difficult for players to fully enjoy the game and access its features. A well-designed and user-friendly interface is essential for an immersive and enjoyable experience.
- Player Skill Matching: Matching players of similar skill levels is important for balanced and competitive gameplay. Inadequate matchmaking algorithms may result in mismatched opponents, leading to one-sided games that are not satisfying for either player. Implementing effective skill-based matchmaking systems can improve the overall game experience.
- Lack of Community Interaction: Online chess games often lack robust community features, limiting opportunities for players to interact, form friendships, and engage in discussions with fellow chess enthusiasts. Incorporating community features such as chat rooms, forums, and social networking elements can enhance player engagement and foster a sense of belonging.
- Inadequate Learning Resources: Some online chess games may lack comprehensive learning resources for players who want to improve their skills. Insufficient tutorials, puzzles, and training materials can make it challenging for beginners to learn the game or for intermediate players to progress to higher levels.

CHAPTER 3

IMPLEMENTATION

4.1 Hardware Requirements:

- **Processor:** Pentium IV or Above
- **RAM :** 2ttB or above
- **Hard Disk :** 50ttB or Above
- **Input Device:** Keyboard, Mouse
- **Output Devices:** Monitor

4.2 Software Requirements:

- **Operating System:** Windows 11
- Brackets
- Basic knowledge about HTML , CSS , JAVASCRIPT

4.3 Implementation Code

Index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <title>Chess Game </title>
   <link rel="stylesheet" href="./style.css">

</head>
```

Online chess game

```
<body>



  <!-- partial:index.partial.html -->
       <div id="game">
       <div class='cellprefix'>8</div>
       <div class='gamecell' id='1_8'></div>
       <div class='gamecell grey' id='2_8'></div>
       <div class='gamecell' id='3_8'></div>
       <div class='gamecell grey' id='4_8'></div>
       <div class='gamecell' id='5_8'></div>
       <div class='gamecell grey' id='6_8'></div>
       <div class='gamecell' id='7_8'></div>
       <div class='gamecell grey' id='8_8'></div><br>
       <div class='cellprefix'>7</div>
       <div class='gamecell grey' id='1_7'></div>
       <div class='gamecell' id='2_7'></div>
       <div class='gamecell grey' id='3_7'></div>
       <div class='gamecell' id='4_7'></div>
       <div class='gamecell grey' id='5_7'></div>
       <div class='gamecell' id='6_7'></div>
       <div class='gamecell grey' id='7_7'></div>
       <div class='gamecell' id='8_7'></div><br>
       <div class='cellprefix'>6</div>
       <div class='gamecell' id='1_6'></div>
       <div class='gamecell grey' id='2_6'></div>
       <div class='gamecell' id='3_6'></div>
       <div class='gamecell grey' id='4_6'></div>
       <div class='gamecell' id='5_6'></div>
       <div class='gamecell grey' id='6_6'></div>
       <div class='gamecell' id='7_6'></div>
       <div class='gamecell grey' id='8_6'></div><br>
       <div class='cellprefix'>5</div>
       <div class='gamecell grey' id='1_5'></div>
       <div class='gamecell' id='2_5'></div>
       <div class='gamecell grey' id='3_5'></div>
       <div class='gamecell' id='4_5'></div>
       <div class='gamecell grey' id='5_5'></div>
       <div class='gamecell' id='6_5'></div>
       <div class='gamecell grey' id='7_5'></div>
       <div class='gamecell' id='8_5'></div><br>
       <div class='cellprefix'>4</div>
       <div class='gamecell' id='1_4'></div>
       <div class='gamecell grey' id='2_4'></div>
       <div class='gamecell' id='3_4'></div>
       <div class='gamecell grey' id='4_4'></div>
       <div class='gamecell' id='5_4'></div>
       <div class='gamecell grey' id='6_4'></div>
       <div class='gamecell' id='7_4'></div>
```

```
        <div class='gamecell grey' id='8_4'></div><br>
        <div class='cellprefix'>3</div>
        <div class='gamecell grey' id='1_3'></div>
        <div class='gamecell' id='2_3'></div>

  <div class='gamecell grey' id='3_3'></div>
        <div class='gamecell' id='4_3'></div>
        <div class='gamecell grey' id='5_3'></div>
        <div class='gamecell' id='6_3'></div>
        <div class='gamecell grey' id='7_3'></div>
        <div class='gamecell' id='8_3'></div><br>
        <div class='cellprefix'>2</div>
        <div class='gamecell' id='1_2'></div>
        <div class='gamecell grey' id='2_2'></div>
        <div class='gamecell' id='3_2'></div>
        <div class='gamecell grey' id='4_2'></div>
        <div class='gamecell' id='5_2'></div>
        <div class='gamecell grey' id='6_2'></div>
        <div class='gamecell' id='7_2'></div>
        <div class='gamecell grey' id='8_2'></div><br>
        <div class='cellprefix'>1</div>
        <div class='gamecell grey' id='1_1'></div>
        <div class='gamecell' id='2_1'></div>
        <div class='gamecell grey' id='3_1'></div>
        <div class='gamecell' id='4_1'></div>
        <div class='gamecell grey' id='5_1'></div>
        <div class='gamecell' id='6_1'></div>
        <div class='gamecell grey' id='7_1'></div>
        <div class='gamecell' id='8_1'></div><br>
        <div class='cellprefix'></div>
        <div class='cellprefix'>a</div>
        <div class='cellprefix'>b</div>
        <div class='cellprefix'>c</div>
        <div class='cellprefix'>d</div>
        <div class='cellprefix'>e</div>
        <div class='cellprefix'>f</div>
        <div class='cellprefix'>g</div>
        <div class='cellprefix'>h</div><br>
        <div id='turn'>It's Whites Turn!</div>
    </div>
    <!-- partial -->
    <script src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js'></script>
    <script src="./script.js"></script>

</body>

</html>
```

Online chess game

Style.css

```
html,
body {
    width: 100%;
    height: 100%;
    padding: 0;
    margin: 0;
    font-family: sans-serif;
}

div * {
    padding: 0;
    margin: 0;
    box-sizing: border-box;
}

#turn {
    max-width: 451px;
    max-height: 30px;
    width: 100%;
    height: 100%;
    position: relative;
    float: right;
    border-radius: 5px;
    border: 1px solid rgb(0, 0, 0);
    border-style: inset;
    text-align: center;
    padding: 5px 0 0 0;
    background: #fff;
    transition: .85s linear;
}

.turnhighlight {
    background: #5cb85c !important;
    color: #fff;
}

#game {
    max-width: 504px;
    max-height: 504px;
    width: 100%;
    height: 100%;
    position: relative;
    margin: 20px auto;
}

.cellprefix {
    width: 100%;
    height: 100%;
    max-width: 50px;
    max-height: 50px;
    float: left;
    margin: 3px;
    padding: 15px 0 0 20px;
}

.gamecell {
    border: 1px solid #000;
    width: 100%;
    height: 100%;
    max-width: 50px;
    max-height: 50px;
    float: left;
    margin: 3px;
    transition: all 0.5s ease-in-out;
    border-radius: 5px;
    font-size: 30pt;
    padding: 0 0 0 6px;
    cursor: pointer;
    z-index: 1;
}



.grey {
    background: rgba(128, 128, 128, 0.801);
}

.green {
    /*background: green !important;*/
    background: rgb(65, 161, 73) !important;
}

.neonblue_txt {
    animation: neonBlueText 1.5s ease-in-out infinite alternate;
}

.neonorange_txt {
    animation: neonOrangeText 1.5s ease-in-out infinite alternate;
}

.neongreen_txt {
    animation: neonGreenText 1.5s ease-in-out infinite alternate;
}

/* --- N E O N    T E X T --- */
@keyframes neonBlueText {
    from {
        text-shadow: 0 0 10px #fff, 0 0 20px #fff, 0 0 30px #fff, 0 0 40px #228dff,
            0 0 70px #228dff, 0 0 80px #228dff, 0 0 100px #228dff, 0 0 150px #228dff;
    }

    to {
        text-shadow: 0 0 5px #fff, 0 0 10px #fff, 0 0 15px #fff, 0 0 20px #228dff,
            0 0 35px #228dff, 0 0 40px #228dff, 0 0 50px #228dff, 0 0 75px #228dff;
    }
}

@keyframes neonOrangeText {
    from {
        text-shadow: 0 0 10px #fff, 0 0 20px #fff, 0 0 30px #fff, 0 0 40px #ff9900,
            0 0 70px #ff9900, 0 0 80px #ff9900, 0 0 100px #ff9900, 0 0 150px #ff9900;
    }

    to {
        text-shadow: 0 0 5px #fff, 0 0 10px #fff, 0 0 15px #fff, 0 0 20px #ff9900,
            0 0 35px #ff9900, 0 0 40px #ff9900, 0 0 50px #ff9900, 0 0 75px #ff9900;
    }
}

@keyframes neonGreenText {
    from {
        text-shadow: 0 0 10px #fff, 0 0 20px #fff, 0 0 30px #fff, 0 0 40px #b6ff00,
            0 0 70px #b6ff00, 0 0 80px #b6ff00, 0 0 100px #b6ff00, 0 0 150px #b6ff00;
    }

    to {
        text-shadow: 0 0 5px #fff, 0 0 10px #fff, 0 0 15px #fff, 0 0 20px #b6ff00,
            0 0 35px #b6ff00, 0 0 40px #b6ff00, 0 0 50px #b6ff00, 0 0 75px #b6ff00;
    }
}

/* --- N E O N --- */

/* animation: neon 1.5s ease-in-out infinite alternate; */

@keyframes neonBlue {
    from {
        box-shadow: 0 0 10px #fff, 0 0 20px #fff, 0 0 30px #fff, 0 0 40px #228dff,
            0 0 70px #228dff, 0 0 80px #228dff, 0 0 100px #228dff, 0 0 150px #228dff;
    }
    to {
        box-shadow: 0 0 5px #fff, 0 0 10px #fff, 0 0 15px #fff, 0 0 20px #228dff,
            0 0 35px #228dff, 0 0 40px #228dff, 0 0 50px #228dff, 0 0 75px #228dff;
    }
}


.shake-little {
    display: inline-block;
    transform-origin: center center;
}

.shake-freeze,
.shake-constant.shake-constant--hover:hover,
.shake-trigger:hover .shake-constant.shake-constant--hover {
    animation-play-state: paused;
}

.shake-freeze:hover,
.shake-trigger:hover .shake-freeze,
.shake-little:hover,
.shake-trigger:hover .shake-little {
    animation-play-state: running;
}

@keyframes shake-little {
    2% {
        transform: translate(1px, 0px) rotate(0.5deg);
    }

    4% {
        transform: translate(1px, 0px) rotate(0.5deg);
    }

    6% {
        transform: translate(1px, 1px) rotate(0.5deg);
    }

    8% {
        transform: translate(0px, 0px) rotate(0.5deg);
    }

    10% {
        transform: translate(1px, 0px) rotate(0.5deg);
    }

    12% {
        transform: translate(0px, 0px) rotate(0.5deg);
}

14% {
   transform: translate(1px, 1px) rotate(0.5deg);
}

16% {
   transform: translate(0px, 1px) rotate(0.5deg);
}

18% {
   transform: translate(1px, 0px) rotate(0.5deg);
}

20% {
   transform: translate(0px, 1px) rotate(0.5deg);
}

22% {
   transform: translate(0px, 0px) rotate(0.5deg);
}

24% {
   transform: translate(0px, 0px) rotate(0.5deg);
}

26% {
   transform: translate(1px, 1px) rotate(0.5deg);
}

28% {
   transform: translate(0px, 1px) rotate(0.5deg);
}

30% {
   transform: translate(0px, 0px) rotate(0.5deg);
}

32% {
   transform: translate(1px, 1px) rotate(0.5deg);
}

34% {
   transform: translate(0px, 1px) rotate(0.5deg);
}

36% {
   transform: translate(0px, 1px) rotate(0.5deg);
}
```

```
38% {
   transform: translate(0px, 0px) rotate(0.5deg);
}

40% {
   transform: translate(1px, 0px) rotate(0.5deg);
}

42% {
   transform: translate(0px, 1px) rotate(0.5deg);
}

44% {
   transform: translate(0px, 1px) rotate(0.5deg);
}

46% {
   transform: translate(0px, 0px) rotate(0.5deg);
}

48% {
   transform: translate(1px, 0px) rotate(0.5deg);
}

50% {
   transform: translate(1px, 1px) rotate(0.5deg);
}

52% {
   transform: translate(0px, 0px) rotate(0.5deg);
}

54% {
   transform: translate(1px, 1px) rotate(0.5deg);
}

56% {
   transform: translate(0px, 1px) rotate(0.5deg);
}

58% {
   transform: translate(1px, 0px) rotate(0.5deg);
}

60% {
   transform: translate(1px, 1px) rotate(0.5deg);
}
```

```
  62% {
     transform: translate(0px, 1px) rotate(0.5deg);
  }

  64% {
     transform: translate(0px, 0px) rotate(0.5deg);
  }

  66% {
     transform: translate(1px, 0px) rotate(0.5deg);
  }

  68% {
     transform: translate(0px, 0px) rotate(0.5deg);
  }

  70% {
     transform: translate(1px, 0px) rotate(0.5deg);
  }

  72% {
     transform: translate(1px, 1px) rotate(0.5deg);
  }

  74% {
     transform: translate(1px, 1px) rotate(0.5deg);
  }

  76% {
     transform: translate(0px, 0px) rotate(0.5deg);
  }

  78% {
     transform: translate(0px, 0px) rotate(0.5deg);
  }

  80% {
     transform: translate(1px, 0px) rotate(0.5deg);
  }

  82% {
     transform: translate(1px, 1px) rotate(0.5deg);
  }

  84% {
     transform: translate(0px, 1px) rotate(0.5deg);
  }

  86% {
        transform: translate(1px, 1px) rotate(0.5deg);
    }

    88% {
        transform: translate(1px, 1px) rotate(0.5deg);
    }

    90% {
        transform: translate(0px, 1px) rotate(0.5deg);
    }

    92% {
        transform: translate(1px, 0px) rotate(0.5deg);
    }

    94% {
        transform: translate(1px, 0px) rotate(0.5deg);
    }

    96% {
        transform: translate(1px, 0px) rotate(0.5deg);
    }

    98% {
        transform: translate(1px, 1px) rotate(0.5deg);
    }

    0%,
    100% {
        transform: translate(0, 0) rotate(0);
    }
}

.shake-little:hover,
.shake-trigger:hover .shake-little,
.shake-little.shake-freeze,
.shake-little.shake-constant {
    animation-name: shake-little;
    animation-duration: 100ms;
    animation-timing-function: ease-in-out;
    animation-iteration-count: infinite;
}
```

Script.js

```
let main = {

    variables: {
        turn: 'w',
        selectedpiece: '',
        highlighted: [],
        pieces: {
            w_king: {
                position: '5_1',
                img: '&#9812;',
                captured: false,
                moved: false,
                type: 'w_king'

            },
            w_queen: {
                position: '4_1',
                img: '&#9813;',
                captured: false,
                moved: false,
                type: 'w_queen'
            },
            w_bishop1: {
                position: '3_1',
                img: '&#9815;',
                captured: false,
                moved: false,
                type: 'w_bishop'
            },
            w_bishop2: {
                position: '6_1',
                img: '&#9815;',
                captured: false,
                moved: false,
                type: 'w_bishop'
            },
            w_knight1: {
                position: '2_1',
                img: '&#9816;',
                captured: false,
                moved: false,
                type: 'w_knight'
            },
            w_knight2: {
                position: '7_1',
                img: '&#9816;',
        captured: false,
        moved: false,
        type: 'w_knight'
    },
    w_rook1: {
        position: '1_1',
        img: '&#9814;',
        captured: false,
        moved: false,
        type: 'w_rook'
    },
    w_rook2: {
        position: '8_1',
        img: '&#9814;',
        captured: false,
        moved: false,
        type: 'w_rook'
    },
    w_pawn1: {
        position: '1_2',
        img: '&#9817;',
        captured: false,
        type: 'w_pawn',
        moved: false
    },
    w_pawn2: {
        position: '2_2',
        img: '&#9817;',
        captured: false,
        type: 'w_pawn',
        moved: false
    },
    w_pawn3: {
        position: '3_2',
        img: '&#9817;',
        captured: false,
        type: 'w_pawn',
        moved: false
    },
    w_pawn4: {
        position: '4_2',
        img: '&#9817;',
        captured: false,
        type: 'w_pawn',
        moved: false
    },
    w_pawn5: {
        position: '5_2',
        img: '&#9817;',
        captured: false,
        type: 'w_pawn',
        moved: false
    },
w_pawn6: {
   position: '6_2',
   img: '&#9817;',
   captured: false,
   type: 'w_pawn',
   moved: false
},
w_pawn7: {
   position: '7_2',
   img: '&#9817;',
   captured: false,
   type: 'w_pawn',
   moved: false
},
w_pawn8: {
   position: '8_2',
   img: '&#9817;',
   captured: false,
   type: 'w_pawn',
   moved: false
},

b_king: {
   position: '5_8',
   img: '&#9818;',
   captured: false,
   moved: false,
   type: 'b_king'
},
b_queen: {
   position: '4_8',
   img: '&#9819;',
   captured: false,
   moved: false,
   type: 'b_queen'
},
b_bishop1: {
   position: '3_8',
   img: '&#9821;',
   captured: false,
   moved: false,
   type: 'b_bishop'
},
b_bishop2: {
   position: '6_8',
   img: '&#9821;',
   captured: false,
   moved: false,
   type: 'b_bishop'
},
b_knight1: {
   position: '2_8',
   img: '&#9822;',
    captured: false,
    moved: false,
    type: 'b_knight'
  },
  b_knight2: {
    position: '7_8',
    img: '&#9822;',
    captured: false,
    moved: false,
    type: 'b_knight'
  },
  b_rook1: {
    position: '1_8',
    img: '&#9820;',
    captured: false,
    moved: false,
    type: 'b_rook'
  },
  b_rook2: {
    position: '8_8',
    img: '&#9820;',
    captured: false,
    moved: false,
    type: 'b_rook'
  },
  b_pawn1: {
    position: '1_7',
    img: '&#9823;',
    captured: false,
    type: 'b_pawn',
    moved: false
  },
  b_pawn2: {
    position: '2_7',
    img: '&#9823;',
    captured: false,
    type: 'b_pawn',
    moved: false
  },
  b_pawn3: {
    position: '3_7',
    img: '&#9823;',
    captured: false,
    type: 'b_pawn',
    moved: false
  },
  b_pawn4: {
    position: '4_7',
    img: '&#9823;',
    captured: false,
    type: 'b_pawn',
    moved: false
  },
        b_pawn5: {
            position: '5_7',
            img: '&#9823;',
            captured: false,
            type: 'b_pawn',
            moved: false
        },
        b_pawn6: {
            position: '6_7',
            img: '&#9823;',
            captured: false,
            type: 'b_pawn',
            moved: false
        },
        b_pawn7: {
            position: '7_7',
            img: '&#9823;',
            captured: false,
            type: 'b_pawn',
            moved: false
        },
        b_pawn8: {
            position: '8_7',
            img: '&#9823;',
            captured: false,
            type: 'b_pawn',
            moved: false
        }

    }
},

methods: {
    gamesetup: function () {
        $('.gamecell').attr('chess', 'null');
        for (let gamepiece in main.variables.pieces) {
            $('#' + main.variables.pieces[gamepiece].position).html(main.variables.pieces[gamepiece].img);
            $('#' + main.variables.pieces[gamepiece].position).attr('chess', gamepiece);
        }
    },

    moveoptions: function (selectedpiece) {

        let position = {
            x: '',
            y: ''
        };
        position.x = main.variables.pieces[selectedpiece].position.split('_')[0];
        position.y = main.variables.pieces[selectedpiece].position.split('_')[1];

        // these options need to be var instead of let
        var options = [];
        var coordinates = [];
```

```
            var startpoint = main.variables.pieces[selectedpiece].position;
            var c1, c2, c3, c4, c5, c6, c7, c8;

            if (main.variables.highlighted.length != 0) {
                main.methods.togglehighlight(main.variables.highlighted);
            }

            switch (main.variables.pieces[selectedpiece].type) {
                case 'w_king':

                    if ($('#6_1').attr('chess') == 'null' && $('#7_1').attr('chess') == 'null' &&
main.variables.pieces['w_king'].moved == false && main.variables.pieces['w_rook2'].moved == false) {
                        coordinates = [{
                            x: 1,
                            y: 1
                        }, {
                            x: 1,
                            y: 0
                        }, {
                            x: 1,
                            y: -1
                        }, {
                            x: 0,
                            y: -1
                        }, {
                            x: -1,
                            y: -1
                        }, {
                            x: -1,
                            y: 0
                        }, {
                            x: -1,
                            y: 1
                        }, {
                            x: 0,
                            y: 1
                        }, {
                            x: 2,
                            y: 0
                        }].map(function (val) {
                            return (parseInt(position.x) + parseInt(val.x)) + '_' + (parseInt(position.y) + parseInt(val.y));
                        });
                    } else {
                        coordinates = [{
                            x: 1,
                            y: 1
                        }, {
                            x: 1,
                            y: 0
                        }, {
                            x: 1,
                            y: -1
                        }, {
```

```
              x: 0,
              y: -1
            }, {
              x: -1,
              y: -1
            }, {
              x: -1,
              y: 0
            }, {
              x: -1,
              y: 1
            }, {
              x: 0,
              y: 1
            }].map(function (val) {
              return (parseInt(position.x) + parseInt(val.x)) + '_' + (parseInt(position.y) + parseInt(val.y));
            });
          }

          options = (main.methods.options(startpoint, coordinates,
main.variables.pieces[selectedpiece].type)).slice(0);
          main.variables.highlighted = options.slice(0);
          main.methods.togglehighlight(options);

          break;
        case 'b_king':

          if ($('#6_8').attr('chess') == 'null' && $('#7_8').attr('chess') == 'null' &&
main.variables.pieces['b_king'].moved == false && main.variables.pieces['b_rook2'].moved == false) {
            coordinates = [{
              x: 1,
              y: 1
            }, {
              x: 1,
              y: 0
            }, {
              x: 1,
              y: -1
            }, {
              x: 0,
              y: -1
            }, {
              x: -1,
              y: -1
            }, {
              x: -1,
              y: 0
            }, {
              x: -1,
              y: 1
            }, {
              x: 0,
              y: 1
```

```
                        }, {
                            x: 2,
                            y: 0
                        }].map(function (val) {
                            return (parseInt(position.x) + parseInt(val.x)) + '_' + (parseInt(position.y) + parseInt(val.y));
                        });
                    } else {
                        coordinates = [{
                            x: 1,
                            y: 1
                        }, {
                            x: 1,
                            y: 0
                        }, {
                            x: 1,
                            y: -1
                        }, {
                            x: 0,
                            y: -1
                        }, {
                            x: -1,
                            y: -1
                        }, {
                            x: -1,
                            y: 0
                        }, {
                            x: -1,
                            y: 1
                        }, {
                            x: 0,
                            y: 1
                        }].map(function (val) {
                            return (parseInt(position.x) + parseInt(val.x)) + '_' + (parseInt(position.y) + parseInt(val.y));
                        });
                    }
                    /*
                      coordinates = [{ x: 1, y: 1 },{ x: 1, y: 0 },{ x: 1, y: -1 },{ x: 0, y: -1 },{ x: -1, y: -1 },{ x: -1, y: 0 },{ x: -1,
y: 1 },{ x: 0, y: 1 }].map(function(val){
                        return (parseInt(position.x) + parseInt(val.x)) + '_' + (parseInt(position.y) + parseInt(val.y));
                      });
                    */
                    options = (main.methods.options(startpoint, coordinates,
main.variables.pieces[selectedpiece].type)).slice(0);
                    main.variables.highlighted = options.slice(0);
                    main.methods.togglehighlight(options);

                    break;
                case 'w_queen':

                    c1 = main.methods.w_options(position, [{
                        x: 1,
                        y: 1
                    }, {
```

```
          x: 2,
          y: 2
        }, {
          x: 3,
          y: 3
        }, {
          x: 4,
          y: 4
        }, {
          x: 5,
          y: 5
        }, {
          x: 6,
          y: 6
        }, {
          x: 7,
          y: 7
        }]);
        c2 = main.methods.w_options(position, [{
          x: 1,
          y: -1
        }, {
          x: 2,
          y: -2
        }, {
          x: 3,
          y: -3
        }, {
          x: 4,
          y: -4
        }, {
          x: 5,
          y: -5
        }, {
          x: 6,
          y: -6
        }, {
          x: 7,
          y: -7
        }]);
        c3 = main.methods.w_options(position, [{
          x: -1,
          y: 1
        }, {
          x: -2,
          y: 2
        }, {
          x: -3,
          y: 3
        }, {
          x: -4,
          y: 4
        }, {
```

```
      x: -5,
      y: 5
    }, {
      x: -6,
      y: 6
    }, {
      x: -7,
      y: 7
    }]);
    c4 = main.methods.w_options(position, [{
      x: -1,
      y: -1
    }, {
      x: -2,
      y: -2
    }, {
      x: -3,
      y: -3
    }, {
      x: -4,
      y: -4
    }, {
      x: -5,
      y: -5
    }, {
      x: -6,
      y: -6
    }, {
      x: -7,
      y: -7
    }]);
    c5 = main.methods.w_options(position, [{
      x: 1,
      y: 0
    }, {
      x: 2,
      y: 0
    }, {
      x: 3,
      y: 0
    }, {
      x: 4,
      y: 0
    }, {
      x: 5,
      y: 0
    }, {
      x: 6,
      y: 0
    }, {
      x: 7,
      y: 0
    }]);
```

```
c6 = main.methods.w_options(position, [{
    x: 0,
    y: 1
}, {
    x: 0,
    y: 2
}, {
    x: 0,
    y: 3
}, {
    x: 0,
    y: 4
}, {
    x: 0,
    y: 5
}, {
    x: 0,
    y: 6
}, {
    x: 0,
    y: 7
}]);
c7 = main.methods.w_options(position, [{
    x: -1,
    y: 0
}, {
    x: -2,
    y: 0
}, {
    x: -3,
    y: 0
}, {
    x: -4,
    y: 0
}, {
    x: -5,
    y: 0
}, {
    x: -6,
    y: 0
}, {
    x: -7,
    y: 0
}]);
c8 = main.methods.w_options(position, [{
    x: 0,
    y: -1
}, {
    x: 0,
    y: -2
}, {
    x: 0,
    y: -3
```

```
        }, {
            x: 0,
            y: -4
        }, {
            x: 0,
            y: -5
        }, {
            x: 0,
            y: -6
        }, {
            x: 0,
            y: -7
        }]);

        coordinates = c1.concat(c2).concat(c3).concat(c4).concat(c5).concat(c6).concat(c7).concat(c8);

        options = coordinates.slice(0);
        main.variables.highlighted = options.slice(0);
        main.methods.togglehighlight(options);

        break;
    case 'b_queen':

        c1 = main.methods.b_options(position, [{
            x: 1,
            y: 1
        }, {
            x: 2,
            y: 2
        }, {
            x: 3,
            y: 3
        }, {
            x: 4,
            y: 4
        }, {
            x: 5,
            y: 5
        }, {
            x: 6,
            y: 6
        }, {
            x: 7,
            y: 7
        }]);
        c2 = main.methods.b_options(position, [{
            x: 1,
            y: -1
        }, {
            x: 2,
            y: -2
        }, {
            x: 3,
        y: -3
    }, {
        x: 4,
        y: -4
    }, {
        x: 5,
        y: -5
    }, {
        x: 6,
        y: -6
    }, {
        x: 7,
        y: -7
    }]);
    c3 = main.methods.b_options(position, [{
        x: -1,
        y: 1
    }, {
        x: -2,
        y: 2
    }, {
        x: -3,
        y: 3
    }, {
        x: -4,
        y: 4
    }, {
        x: -5,
        y: 5
    }, {
        x: -6,
        y: 6
    }, {
        x: -7,
        y: 7
    }]);
    c4 = main.methods.b_options(position, [{
        x: -1,
        y: -1
    }, {
        x: -2,
        y: -2
    }, {
        x: -3,
        y: -3
    }, {
        x: -4,
        y: -4
    }, {
        x: -5,
        y: -5
    }, {
        x: -6,
        y: -6
    }, {
        x: -7,
        y: -7
    }]);
    c5 = main.methods.b_options(position, [{
        x: 1,
        y: 0
    }, {
        x: 2,
        y: 0
    }, {
        x: 3,
        y: 0
    }, {
        x: 4,
        y: 0
    }, {
        x: 5,
        y: 0
    }, {
        x: 6,
        y: 0
    }, {
        x: 7,
        y: 0
    }]);
    c6 = main.methods.b_options(position, [{
        x: 0,
        y: 1
    }, {
        x: 0,
        y: 2
    }, {
        x: 0,
        y: 3
    }, {
        x: 0,
        y: 4
    }, {
        x: 0,
        y: 5
    }, {
        x: 0,
        y: 6
    }, {
        x: 0,
        y: 7
    }]);
    c7 = main.methods.b_options(position, [{
        x: -1,
        y: 0
    }, {
            x: -2,
            y: 0
        }, {
            x: -3,
            y: 0
        }, {
            x: -4,
            y: 0
        }, {
            x: -5,
            y: 0
        }, {
            x: -6,
            y: 0
        }, {
            x: -7,
            y: 0
        }]);
        c8 = main.methods.b_options(position, [{
            x: 0,
            y: -1
        }, {
            x: 0,
            y: -2
        }, {
            x: 0,
            y: -3
        }, {
            x: 0,
            y: -4
        }, {
            x: 0,
            y: -5
        }, {
            x: 0,
            y: -6
        }, {
            x: 0,
            y: -7
        }]);

        coordinates = c1.concat(c2).concat(c3).concat(c4).concat(c5).concat(c6).concat(c7).concat(c8);

        options = coordinates.slice(0);
        main.variables.highlighted = options.slice(0);
        main.methods.togglehighlight(options);

        break;

    case 'w_bishop':

        c1 = main.methods.w_options(position, [{
            x: 1,
    y: 1
}, {
    x: 2,
    y: 2
}, {
    x: 3,
    y: 3
}, {
    x: 4,
    y: 4
}, {
    x: 5,
    y: 5
}, {
    x: 6,
    y: 6
}, {
    x: 7,
    y: 7
}]);
c2 = main.methods.w_options(position, [{
    x: 1,
    y: -1
}, {
    x: 2,
    y: -2
}, {
    x: 3,
    y: -3
}, {
    x: 4,
    y: -4
}, {
    x: 5,
    y: -5
}, {
    x: 6,
    y: -6
}, {
    x: 7,
    y: -7
}]);
c3 = main.methods.w_options(position, [{
    x: -1,
    y: 1
}, {
    x: -2,
    y: 2
}, {
    x: -3,
    y: 3
}, {
    x: -4,
        y: 4
      }, {
        x: -5,
        y: 5
      }, {
        x: -6,
        y: 6
      }, {
        x: -7,
        y: 7
      }]);
      c4 = main.methods.w_options(position, [{
        x: -1,
        y: -1
      }, {
        x: -2,
        y: -2
      }, {
        x: -3,
        y: -3
      }, {
        x: -4,
        y: -4
      }, {
        x: -5,
        y: -5
      }, {
        x: -6,
        y: -6
      }, {
        x: -7,
        y: -7
      }]);

      coordinates = c1.concat(c2).concat(c3).concat(c4);

      options = coordinates.slice(0);
      main.variables.highlighted = options.slice(0);
      main.methods.togglehighlight(options);

      break;

    case 'b_bishop':

      c1 = main.methods.b_options(position, [{
        x: 1,
        y: 1
      }, {
        x: 2,
        y: 2
      }, {
        x: 3,
        y: 3
```

```
}, {
    x: 4,
    y: 4
}, {
    x: 5,
    y: 5
}, {
    x: 6,
    y: 6
}, {
    x: 7,
    y: 7
}]);
c2 = main.methods.b_options(position, [{
    x: 1,
    y: -1
}, {
    x: 2,
    y: -2
}, {
    x: 3,
    y: -3
}, {
    x: 4,
    y: -4
}, {
    x: 5,
    y: -5
}, {
    x: 6,
    y: -6
}, {
    x: 7,
    y: -7
}]);
c3 = main.methods.b_options(position, [{
    x: -1,
    y: 1
}, {
    x: -2,
    y: 2
}, {
    x: -3,
    y: 3
}, {
    x: -4,
    y: 4
}, {
    x: -5,
    y: 5
}, {
    x: -6,
    y: 6
```

```
          }, {
            x: -7,
            y: 7
          }]);
          c4 = main.methods.b_options(position, [{
            x: -1,
            y: -1
          }, {
            x: -2,
            y: -2
          }, {
            x: -3,
            y: -3
          }, {
            x: -4,
            y: -4
          }, {
            x: -5,
            y: -5
          }, {
            x: -6,
            y: -6
          }, {
            x: -7,
            y: -7
          }]);

          coordinates = c1.concat(c2).concat(c3).concat(c4);

          options = coordinates.slice(0);
          main.variables.highlighted = options.slice(0);
          main.methods.togglehighlight(options);
          break;
        case 'w_knight':

          coordinates = [{
            x: -1,
            y: 2
          }, {
            x: 1,
            y: 2
          }, {
            x: 1,
            y: -2
          }, {
            x: -1,
            y: -2
          }, {
            x: 2,
            y: 1
          }, {
            x: 2,
            y: -1
```

```
                    }, {
                        x: -2,
                        y: -1
                    }, {
                        x: -2,
                        y: 1
                    }].map(function (val) {
                        return (parseInt(position.x) + parseInt(val.x)) + '_' + (parseInt(position.y) + parseInt(val.y));
                    });

                    options = (main.methods.options(startpoint, coordinates,
main.variables.pieces[selectedpiece].type)).slice(0);
                    main.variables.highlighted = options.slice(0);
                    main.methods.togglehighlight(options);

                    break;
                case 'b_knight':

                    coordinates = [{
                        x: -1,
                        y: 2
                    }, {
                        x: 1,
                        y: 2
                    }, {
                        x: 1,
                        y: -2
                    }, {
                        x: -1,
                        y: -2
                    }, {
                        x: 2,
                        y: 1
                    }, {
                        x: 2,
                        y: -1
                    }, {
                        x: -2,
                        y: -1
                    }, {
                        x: -2,
                        y: 1
                    }].map(function (val) {
                        return (parseInt(position.x) + parseInt(val.x)) + '_' + (parseInt(position.y) + parseInt(val.y));
                    });

                    options = (main.methods.options(startpoint, coordinates,
main.variables.pieces[selectedpiece].type)).slice(0);
                    main.variables.highlighted = options.slice(0);
                    main.methods.togglehighlight(options);

                    break;
                case 'w_rook':
```

```
c1 = main.methods.w_options(position, [{
   x: 1,
   y: 0
}, {
   x: 2,
   y: 0
}, {
   x: 3,
   y: 0
}, {
   x: 4,
   y: 0
}, {
   x: 5,
   y: 0
}, {
   x: 6,
   y: 0
}, {
   x: 7,
   y: 0
}]);
c2 = main.methods.w_options(position, [{
   x: 0,
   y: 1
}, {
   x: 0,
   y: 2
}, {
   x: 0,
   y: 3
}, {
   x: 0,
   y: 4
}, {
   x: 0,
   y: 5
}, {
   x: 0,
   y: 6
}, {
   x: 0,
   y: 7
}]);
c3 = main.methods.w_options(position, [{
   x: -1,
   y: 0
}, {
   x: -2,
   y: 0
}, {
   x: -3,
            y: 0
        }, {
            x: -4,
            y: 0
        }, {
            x: -5,
            y: 0
        }, {
            x: -6,
            y: 0
        }, {
            x: -7,
            y: 0
        }]);
        c4 = main.methods.w_options(position, [{
            x: 0,
            y: -1
        }, {
            x: 0,
            y: -2
        }, {
            x: 0,
            y: -3
        }, {
            x: 0,
            y: -4
        }, {
            x: 0,
            y: -5
        }, {
            x: 0,
            y: -6
        }, {
            x: 0,
            y: -7
        }]);

        coordinates = c1.concat(c2).concat(c3).concat(c4);

        options = coordinates.slice(0);
        main.variables.highlighted = options.slice(0);
        main.methods.togglehighlight(options);

        break;
    case 'b_rook':

        c1 = main.methods.b_options(position, [{
            x: 1,
            y: 0
        }, {
            x: 2,
            y: 0
        }, {
```

```
    x: 3,
    y: 0
}, {
    x: 4,
    y: 0
}, {
    x: 5,
    y: 0
}, {
    x: 6,
    y: 0
}, {
    x: 7,
    y: 0
}]);
c2 = main.methods.b_options(position, [{
    x: 0,
    y: 1
}, {
    x: 0,
    y: 2
}, {
    x: 0,
    y: 3
}, {
    x: 0,
    y: 4
}, {
    x: 0,
    y: 5
}, {
    x: 0,
    y: 6
}, {
    x: 0,
    y: 7
}]);
c3 = main.methods.b_options(position, [{
    x: -1,
    y: 0
}, {
    x: -2,
    y: 0
}, {
    x: -3,
    y: 0
}, {
    x: -4,
    y: 0
}, {
    x: -5,
    y: 0
}, {
```

```
            x: -6,
            y: 0
        }, {
            x: -7,
            y: 0
        }]);
        c4 = main.methods.b_options(position, [{
            x: 0,
            y: -1
        }, {
            x: 0,
            y: -2
        }, {
            x: 0,
            y: -3
        }, {
            x: 0,
            y: -4
        }, {
            x: 0,
            y: -5
        }, {
            x: 0,
            y: -6
        }, {
            x: 0,
            y: -7
        }]);

        coordinates = c1.concat(c2).concat(c3).concat(c4);

        options = coordinates.slice(0);
        main.variables.highlighted = options.slice(0);
        main.methods.togglehighlight(options);

        break;
    case 'w_pawn':

        if (main.variables.pieces[selectedpiece].moved == false) {

            coordinates = [{
                x: 0,
                y: 1
            }, {
                x: 0,
                y: 2
            }, {
                x: 1,
                y: 1
            }, {
                x: -1,
                y: 1
            }].map(function (val) {
```

```
                            return (parseInt(position.x) + parseInt(val.x)) + '_' + (parseInt(position.y) + parseInt(val.y));
                        });

                    } else if (main.variables.pieces[selectedpiece].moved == true) {

                        coordinates = [{
                            x: 0,
                            y: 1
                        }, {
                            x: 1,
                            y: 1
                        }, {
                            x: -1,
                            y: 1
                        }].map(function (val) {
                            return (parseInt(position.x) + parseInt(val.x)) + '_' + (parseInt(position.y) + parseInt(val.y));
                        });

                    }

                    options = (main.methods.options(startpoint, coordinates,
main.variables.pieces[selectedpiece].type)).slice(0);
                    main.variables.highlighted = options.slice(0);
                    main.methods.togglehighlight(options);

                    break;

                case 'b_pawn':

                    // calculate pawn options
                    if (main.variables.pieces[selectedpiece].moved == false) {

                        coordinates = [{
                            x: 0,
                            y: -1
                        }, {
                            x: 0,
                            y: -2
                        }, {
                            x: 1,
                            y: -1
                        }, {
                            x: -1,
                            y: -1
                        }].map(function (val) {
                            return (parseInt(position.x) + parseInt(val.x)) + '_' + (parseInt(position.y) + parseInt(val.y));
                        });

                    } else if (main.variables.pieces[selectedpiece].moved == true) {

                        coordinates = [{
                            x: 0,
                            y: -1
                        }, {
                            x: 1,
                            y: -1
                        }, {
                            x: -1,
                            y: -1
                        }].map(function (val) {
                            return (parseInt(position.x) + parseInt(val.x)) + '_' + (parseInt(position.y) + parseInt(val.y));
                        });

                    }

                    options = (main.methods.options(startpoint, coordinates, main.variables.pieces[selectedpiece].type)).slice(0);
                    main.variables.highlighted = options.slice(0);
                    main.methods.togglehighlight(options);

                    break;

            }
        },

        options: function (startpoint, coordinates, piecetype) { // first check if any of the possible coordinates is out of bounds;

            coordinates = coordinates.filter(val => {
                let pos = {
                    x: 0,
                    y: 0
                };
                pos.x = parseInt(val.split('_')[0]);
                pos.y = parseInt(val.split('_')[1]);

                if (!(pos.x < 1) && !(pos.x > 8) && !(pos.y < 1) && !(pos.y > 8)) { // if it is not out of bounds, return the coordinate;
                    return val;
                }
            });

            switch (piecetype) {

                case 'w_king':

                    coordinates = coordinates.filter(val => {
                        return ($('#' + val).attr('chess') == 'null' || ($('#' + val).attr('chess')).slice(0, 1) == 'b');
                    });

                    break;
                case 'b_king':

                    coordinates = coordinates.filter(val => {
                        return ($('#' + val).attr('chess') == 'null' || ($('#' + val).attr('chess')).slice(0, 1) == 'w');
                    });
                    break;
                case 'w_knight':

                    coordinates = coordinates.filter(val => {
                        return ($('#' + val).attr('chess') == 'null' || ($('#' + val).attr('chess')).slice(0, 1) == 'b');
                    });

                    break;

                case 'b_knight':

                    coordinates = coordinates.filter(val => {
                        return ($('#' + val).attr('chess') == 'null' || ($('#' + val).attr('chess')).slice(0, 1) == 'w');
                    });

                    break;

                case 'w_pawn':

                    coordinates = coordinates.filter(val => {
                        let sp = {
                            x: 0,
                            y: 0
                        };
                        let coordinate = val.split('_');

                        sp.x = startpoint.split('_')[0];
                        sp.y = startpoint.split('_')[1];

                        if (coordinate[0] < sp.x || coordinate[0] > sp.x) { // if the coordinate is on either side of the center, check if it has an opponent piece on it;
                            return ($('#' + val).attr('chess') != 'null' && ($('#' + val).attr('chess')).slice(0, 1) == 'b'); // return coordinates with opponent pieces on them
                        } else { // else if the coordinate is in the center;
                            if (coordinate[1] == (parseInt(sp.y) + 2) && $('#' + sp.x + '_' + (parseInt(sp.y) + 1)).attr('chess') != 'null') {
                                // do nothing if this is the pawns first move, and there is a piece in front of the 2nd coordinate;
                            } else {
                                return ($('#' + val).attr('chess') == 'null'); // otherwise return the coordinate if there is no chess piece on it;
                            }
                        }

                    });

                    break;

                case 'b_pawn':

                    coordinates = coordinates.filter(val => {
                        let sp = {
                            x: 0,
                            y: 0
                        };
                        let coordinate = val.split('_');

                        sp.x = startpoint.split('_')[0];
                        sp.y = startpoint.split('_')[1];

                        if (coordinate[0] < sp.x || coordinate[0] > sp.x) { // if the coordinate is on either side of the center, check if it has an opponent piece on it;
                            return ($('#' + val).attr('chess') != 'null' && ($('#' + val).attr('chess')).slice(0, 1) == 'w'); // return coordinates with opponent pieces on them
                        } else { // else if the coordinate is in the center;
                            if (coordinate[1] == (parseInt(sp.y) - 2) && $('#' + sp.x + '_' + (parseInt(sp.y) - 1)).attr('chess') != 'null') {
                                // do nothing if this is the pawns first move, and there is a piece in front of the 2nd coordinate;
                            } else {
                                return ($('#' + val).attr('chess') == 'null'); // otherwise return the coordinate if there is no chess piece on it;
                            }
                        }
                    });

                    break;
            }

            return coordinates;
        },

        w_options: function (position, coordinates) {

            let flag = false;

            coordinates = coordinates.map(function (val) { // convert the x,y into actual grid id coordinates;
                return (parseInt(position.x) + parseInt(val.x)) + '_' + (parseInt(position.y) + parseInt(val.y));
            }).filter(val => {
                let pos = {
                    x: 0,
                    y: 0
                };
                pos.x = parseInt(val.split('_')[0]);
                pos.y = parseInt(val.split('_')[1]);

                if (!(pos.x < 1) && !(pos.x > 8) && !(pos.y < 1) && !(pos.y > 8)) { // if it is not out of bounds, return the coordinate;
                    return val;
                }
            }).filter(val => { // algorithm to determine line-of-sight movement options for bishop/rook/queen;
                if (flag == false) {
                    if ($('#' + val).attr('chess') == 'null') {
                        console.log(val)
                        return val;
                    } else if (($('#' + val).attr('chess')).slice(0, 1) == 'b') {
                        flag = true;
                        console.log(val)
                        return val;
                    } else if (($('#' + val).attr('chess')).slice(0, 1) == 'w') {
                        console.log(val + '-3')
                        flag = true;
                    }
                }
            });

            return coordinates;

        },

        b_options: function (position, coordinates) {

            let flag = false;

            coordinates = coordinates.map(function (val) { // convert the x,y into actual grid id coordinates;
                return (parseInt(position.x) + parseInt(val.x)) + '_' + (parseInt(position.y) + parseInt(val.y));
            }).filter(val => {
                let pos = {
                    x: 0,
                    y: 0
                };
                pos.x = parseInt(val.split('_')[0]);
                pos.y = parseInt(val.split('_')[1]);

                if (!(pos.x < 1) && !(pos.x > 8) && !(pos.y < 1) && !(pos.y > 8)) { // if it is not out of bounds, return the
        coordinate;
                    return val;
                }
            }).filter(val => { // algorithm to determine line-of-sight movement options for bishop/rook/queen;
                if (flag == false) {
                    if ($('#' + val).attr('chess') == 'null') {
                        return val;
                    } else if (($('#' + val).attr('chess')).slice(0, 1) == 'w') {
                        flag = true;
                        return val;
                    } else if (($('#' + val).attr('chess')).slice(0, 1) == 'b') {
                        flag = true;
                    }
                }
            });

            return coordinates;

        },

        capture: function (target) {
            let selectedpiece = {
                name: $('#' + main.variables.selectedpiece).attr('chess'),
                id: main.variables.selectedpiece
            };
            //new cell
            $('#' + target.id).html(main.variables.pieces[selectedpiece.name].img);
            $('#' + target.id).attr('chess', selectedpiece.name);
            //old cell
            $('#' + selectedpiece.id).html('');
            $('#' + selectedpiece.id).attr('chess', 'null');
            //moved piece
            main.variables.pieces[selectedpiece.name].position = target.id;
            main.variables.pieces[selectedpiece.name].moved = true;
            // captured piece
            main.variables.pieces[target.name].captured = true;
            /*
            // toggle highlighted coordinates
            main.methods.togglehighlight(main.variables.highlighted);
            main.variables.highlighted.length = 0;
            // set the selected piece to '' again
            main.variables.selectedpiece = '';
            */

        },

        move: function (target) {

            let selectedpiece = $('#' + main.variables.selectedpiece).attr('chess');

            // new cell
            $('#' + target.id).html(main.variables.pieces[selectedpiece].img);
            $('#' + target.id).attr('chess', selectedpiece);
            // old cell
            $('#' + main.variables.selectedpiece).html('');
            $('#' + main.variables.selectedpiece).attr('chess', 'null');
            main.variables.pieces[selectedpiece].position = target.id;
            main.variables.pieces[selectedpiece].moved = true;

            /*
            // toggle highlighted coordinates
            main.methods.togglehighlight(main.variables.highlighted);
            main.variables.highlighted.length = 0;
            // set the selected piece to '' again
            main.variables.selectedpiece = '';
            */
        },

        endturn: function () {

            if (main.variables.turn == 'w') {
                main.variables.turn = 'b';

                // toggle highlighted coordinates
                main.methods.togglehighlight(main.variables.highlighted);
                main.variables.highlighted.length = 0;
```

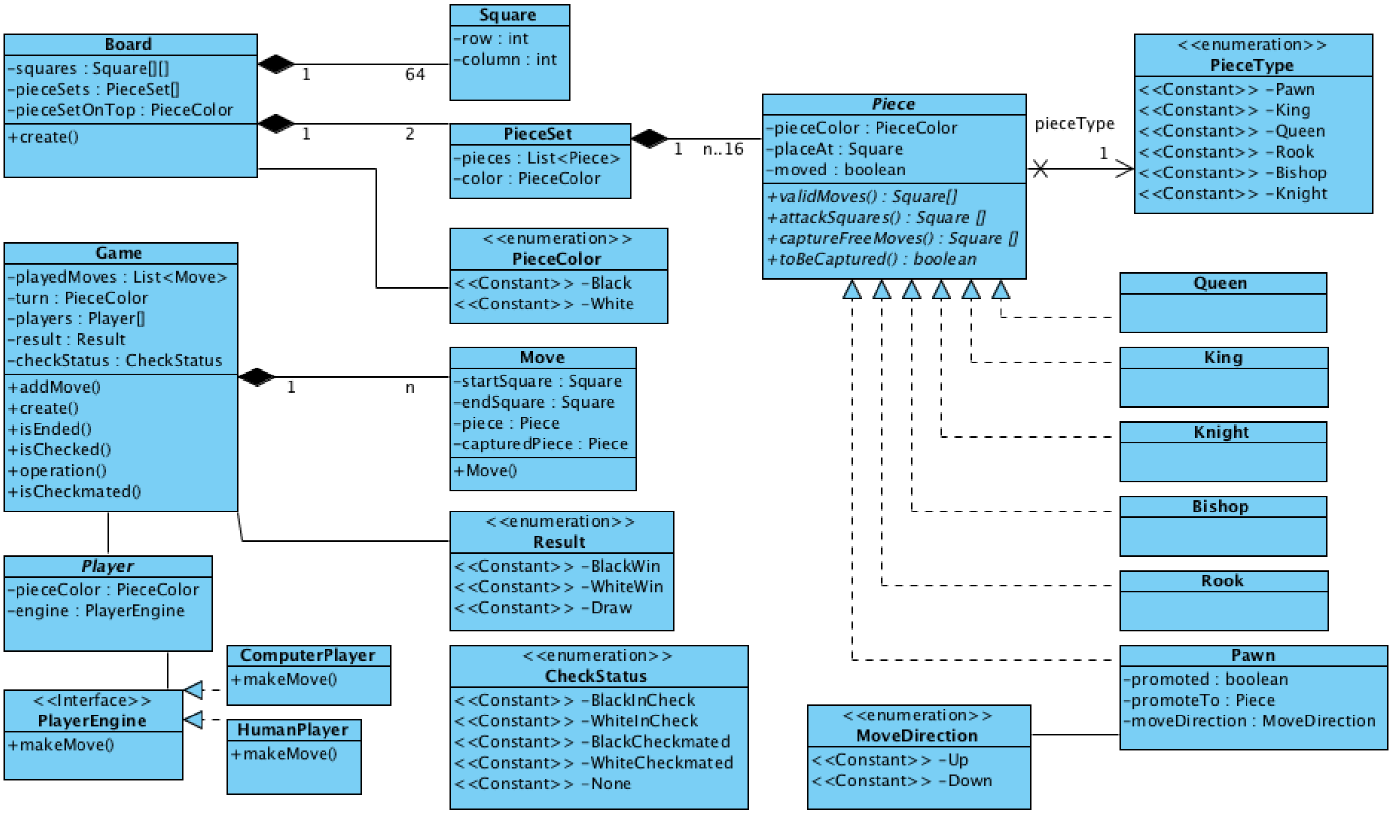
```
                // set the selected piece to " again
                main.variables.selectedpiece = ";

                $('#turn').html("It's Blacks Turn");

                $('#turn').addClass('turnhighlight');
                window.setTimeout(function () {
                    $('#turn').removeClass('turnhighlight');
                }, 1500);

            } else if (main.variables.turn = 'b') {
                main.variables.turn = 'w';

                // toggle highlighted coordinates
                main.methods.togglehighlight(main.variables.highlighted);
                main.variables.highlighted.length = 0;
                // set the selected piece to " again
                main.variables.selectedpiece = ";

                $('#turn').html("It's Whites Turn");

                $('#turn').addClass('turnhighlight');
                window.setTimeout(function () {
                    $('#turn').removeClass('turnhighlight');
                }, 1500);

            }

        },

        togglehighlight: function (options) {
            options.forEach(function (element, index, array) {
                $('#' + element).toggleClass("green shake-little neongreen_txt");
            });
        },

    }
};

$(document).ready(function () {
    main.methods.gamesetup();

    $('.gamecell').click(function (e) {

        var selectedpiece = {
            name: ",
            id: main.variables.selectedpiece
        };

        if (main.variables.selectedpiece == ") {
            selectedpiece.name = $('#' + e.target.id).attr('chess');
        } else {
            selectedpiece.name = $('#' + main.variables.selectedpiece).attr('chess');
```

```
}

var target = {
    name: $(this).attr('chess'),
    id: e.target.id
};

if (main.variables.selectedpiece == '' && target.name.slice(0, 1) == main.variables.turn) { // show options

    // moveoptions
    main.variables.selectedpiece = e.target.id;
    main.methods.moveoptions($(this).attr('chess'));

} else if (main.variables.selectedpiece != '' && target.name == 'null') { // move selected piece piece

    if (selectedpiece.name == 'w_king' || selectedpiece.name == 'b_king') {

        let t0 = (selectedpiece.name = 'w_king');
        let t1 = (selectedpiece.name = 'b_king');
        let t2 = (main.variables.pieces[selectedpiece.name].moved == false);
        let t3 = (main.variables.pieces['b_rook2'].moved == false);
        let t4 = (main.variables.pieces['w_rook2'].moved == false);
        let t5 = (target.id == '7_8');
        let t6 = (target.id == '7_1');

        if (t0 && t2 && t4 && t6) { // castle w_king

            let k_position = '5_1';
            let k_target = '7_1';
            let r_position = '8_1';
            let r_target = '6_1';

            main.variables.pieces['w_king'].position = '7_1';
            main.variables.pieces['w_king'].moved = true;
            $('#' + k_position).html('');
            $('#' + k_position).attr('chess', 'null');
            $('#' + k_target).html(main.variables.pieces['w_king'].img);
            $('#' + k_target).attr('chess', 'w_king');

            main.variables.pieces['w_rook2'].position = '6_1';
            main.variables.pieces['w_rook2'].moved = true;
            $('#' + r_position).html('');
            $('#' + r_position).attr('chess', 'null');
            $('#' + r_target).html(main.variables.pieces['w_rook2'].img);
            $('#' + r_target).attr('chess', 'w_rook2');

            main.methods.endturn();

        } else if (t1 && t2 && t3 && t5) { // castle b_king

            let k_position = '5_8';
            let k_target = '7_8';
            let r_position = '8_8';
```

```
                 let r_target = '6_8';

                    // w_king
                    main.variables.pieces['b_king'].position = '7_8';
                    main.variables.pieces['b_king'].moved = true;
                    $('#' + k_position).html('');
                    $('#' + k_position).attr('chess', 'null');
                    $('#' + k_target).html(main.variables.pieces['b_king'].img);
                    $('#' + k_target).attr('chess', 'b_king');

                    main.variables.pieces['b_rook2'].position = '6_8';
                    main.variables.pieces['b_rook2'].moved = true;
                    $('#' + r_position).html('');
                    $('#' + r_position).attr('chess', 'null');
                    $('#' + r_target).html(main.variables.pieces['b_rook2'].img);
                    $('#' + r_target).attr('chess', 'b_rook2');

                    main.methods.endturn();

                } else { // move selectedpiece
                    main.methods.move(target);
                    main.methods.endturn();
                }

            } else { // else if selecedpiece.name is not white/black king than move

                main.methods.move(target);
                main.methods.endturn();

            }

        } else if (main.variables.selectedpiece != '' && target.name != 'null' && target.id != selectedpiece.id &&
selectedpiece.name.slice(0, 1) != target.name.slice(0, 1)) { // capture a piece

            if (selectedpiece.id != target.id && main.variables.highlighted.indexOf(target.id) != (-1)) { // if it's not trying to
capture pieces not in its movement range

                // capture
                main.methods.capture(target)
                main.methods.endturn();

            }

        } else if (main.variables.selectedpiece != '' && target.name != 'null' && target.id != selectedpiece.id &&
selectedpiece.name.slice(0, 1) == target.name.slice(0, 1)) { // toggle move options

            // toggle
            main.methods.togglehighlight(main.variables.highlighted);
            main.variables.highlighted.length = 0;

            main.variables.selectedpiece = target.id;
```

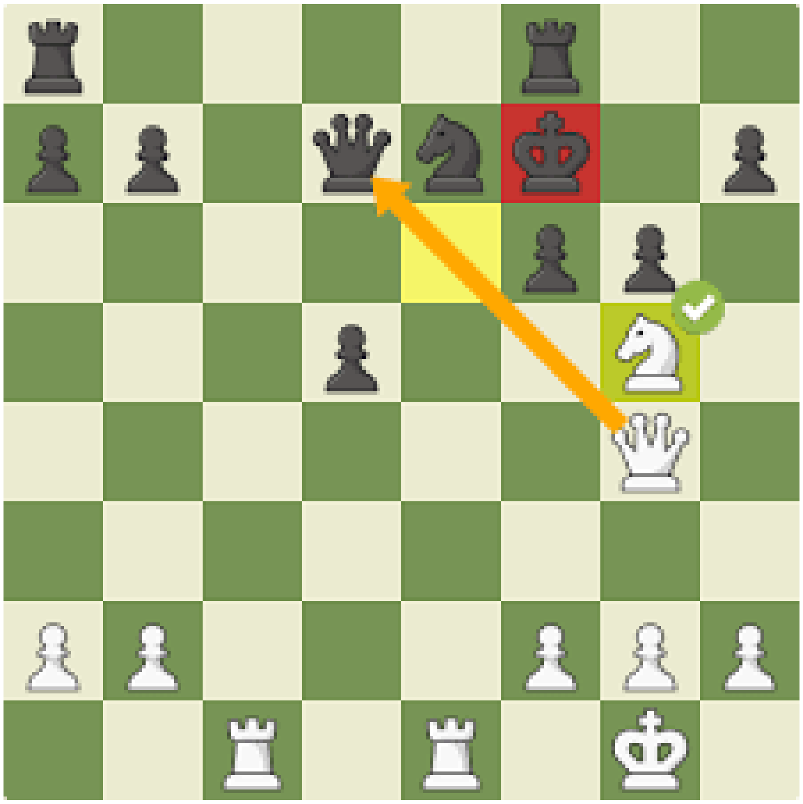
CHAPTER 4

SYSTEM ARCHITECTURE



CHAPTER 5

RESULT



CHAPTER 6

APPLICATIONS & ADVANTAGES AND DISADVANTAGES

6.1 APPLICATIONS

- Online chess games provide a platform for players to develop and enhance their chess skills
- Online chess platforms also host official tournaments and championships, providing opportunities for players to showcase their abilities and compete for recognition.

6.2 ADVANTAGES & DISADVANTAGES

6.2.1ADVANTAGES

1. Chess is a highly strategic and complex game that engages the mind. Regularly playing chess can improve critical thinking, problem-solving, and analytical skills. It requires players to anticipate moves, plan ahead, and consider various possibilities, stimulating the brain and enhancing cognitive abilities.
2. involves remembering and recalling various positions, strategies, and patterns. The game challenges players to remember past moves, anticipate future moves, and recognize patterns on the board. Regular chess practice can improve memory retention and recall abilities.
3. Chess helps develop strategic thinking, foresight, and the ability to make calculated moves
4. Players can continuously learn and improve their skills, making chess a lifelong pursuit of knowledge and self-improvement.
5. Online chess games create a vibrant community where players can connect, communicate, and interact with fellow chess enthusiasts from around the world. Players can join chess clubs, participate in forums, engage in friendly matches, and share their experiences and knowledge. This social aspect of online chess fosters a sense of belonging and camaraderie among players.

6.2.2 DISADVANTAGES

1. Network Connectivity Issues: Online chess games heavily rely on stable internet connections. Any disruptions in the network, such as high latency, frequent disconnections, or slow response times, can negatively impact the gameplay experience. These issues can lead to frustration, unfair advantages, and interruptions during critical moments of the game.

2. Cheating and Fair Play Concerns: Online chess games face challenges related to cheating and fair play. Players can use computer assistance, refer to chess engines, or engage in other unethical practices to gain an unfair advantage. Ensuring fair play and implementing effective anti-cheating measures can be challenging, as new cheating techniques constantly emerge.

3. Lack of Physical Presence and Human Interaction: Online chess games lack the physical presence and human interaction that traditional over-the-board chess offers. Some players prefer the tactile experience of moving physical chess pieces and the face-to-face interaction with their opponents. Online chess may feel impersonal and can limit the social and communal aspects of the game.

4. Loss of Tradition and Historical Significance: Online chess games, while convenient and accessible, may diminish the traditional and historical significance associated with the game. The cultural and historical elements that surround physical chess sets, tournaments, and chess clubs may not be fully replicated in the online environment, leading to a loss of some of the traditional aspects of chess.

CONCLUSION

In conclusion, online chess games have become immensely popular and have revolutionized the way people engage with this ancient game. They offer a range of advantages and benefits, including mental stimulation, strategic thinking, cognitive development, and social interaction. Online chess games provide a platform for competitive gaming, chess education, remote learning, and coaching, as well as analysis and research. They cater to both casual players seeking entertainment and serious players aiming to improve their skills. The future of online chess games holds even more exciting possibilities, including virtual reality and augmented reality experiences, enhanced AI integration, cross-platform compatibility, advanced spectator features, and the potential for eSports and a competitive chess scene. Personalization, customization, and improved training resources are also expected to enrich the online chess gaming experience. Furthermore, the integration of blockchain technology could introduce new dimensions of security, ownership, and asset trading.

REFERENCES

https://www.geeksforgeeks.org

www.wikipedia.com