Department of Computer Science and Engineering

Course Code: CSE-3636
Course Title: Artificial Intelligence Lab
Project Title: AI based Chess Game using Python

Project Documentation

Week 7: Castling

Submitted by

Lamisa Mashiat (C201249)

Sohana Tasneem (C201266)

Sadia Haque Chowdhury (C201270)

Submitted to

Subrina Akter
Assistant Professor
Dept. of CSE

29th April, 2023

The updated version of code, which now includes the highly anticipated castling move. Castling is a powerful and strategic maneuver that adds an exciting new dimension to the game of chess. With this update, players can now enjoy a more realistic and challenging game of chess, as they try to outmaneuver their opponents and secure victory.

Here's a list of the updates that we did this week:

- **castling_rights** (dict): A dictionary that stores the castling rights for both players. It has four keys: **"w_k"**, **"w_q"**, **"b_k"**, and **"b_q"**, which represent the right to castle kingside (**"k"**) or queenside (**"q"**) for white (**"w"**) or black (**"b"**) respectively. The value of each key is a boolean that indicates whether the player has the right to castle in that direction.

- **New Class: CastleRights()** is a new class created to keep track of the current castling rights of both players. It takes four boolean arguments that represent the current castling rights of each player for kingside and queenside castling.

- **Initialization:** The **__init__()** method of the **Board** class is updated to initialize **self.currentCastlingRight** as a new **CastleRights** object with all castling rights enabled, and a new list **self.castleRightsLog** to keep track of changes to castling rights.

- **Update Castling Rights:** The **updateCastleRights()** method is updated to disable castling rights of the player whose piece is moved, and also any rook that is moved.

- **Make Move:** The **makeMove()** method is updated to handle castling moves. If a move is a castling move, it moves the rook and updates the board accordingly. It also updates the current castling rights and adds a new **CastleRights** object to **self.castleRightsLog**.

- **Undo Move:** The **undoMove()** method is updated to undo castling moves and restore the previous castling rights from **self.castleRightsLog**.

- **New Methods:** Two new methods **getKingsideCastleMoves()** and **getQueensideCastleMoves()** are added to generate all valid castle moves for the king at a given position. These methods check if the squares between the king and rook are empty and not under attack, and add a new **Move** object with **isCastleMove=True** to the **moves** list.

- **Update getValidMoves():** The **getValidMoves()** method is updated to call the new **getCastleMoves()** method for the king's position if it is a valid move.