Hier eine Zusammenfassung der Implementierung des `ChessApp`-Programms mit den Codezeilen und dem Umfang jeder Funktion, die die Hauptlogik und die Benutzeroberfläche des Schachspiels enthält. Jede Funktion ist so gestaltet, dass sie spezifische Aufgaben übernimmt, die entweder die Funktionalität der Anwendung erweitern oder die Interaktivität der Benutzeroberfläche verbessern. Die Informationen zu den Funktionen umfassen die Zeilennummern sowie die Anzahl der Codezeilen, um eine bessere Übersicht über die Struktur und den Umfang des Projekts zu ermöglichen.

1. \*\*`\_\_init\_\_` (Zeilen 22-87, 66 Zeilen)\*\*: Die `\_\_init\_\_`-Methode initialisiert die Hauptkomponenten der Klasse `ChessApp`, darunter die Erstellung der Benutzeroberfläche, die Einstellung der grundlegenden Spielmechanik und die Konfiguration des Schachbretts. Diese Methode stellt sicher, dass alle benötigten Variablen vorbereitet und alle Module geladen sind.

2. \*\*`toggle\_sidebar` (Zeilen 89-101, 13 Zeilen)\*\*: Diese Methode ermöglicht das Umschalten einer Seitenleiste in der Anwendung, die zusätzliche Optionen wie "Schachregeln" und "Dokumentation" enthält. Die Methode erstellt oder zerstört die Seitenleiste basierend auf ihrem aktuellen Zustand.

3. \*\*`show\_play\_options` (Zeilen 103-113, 11 Zeilen)\*\*: Zeigt dem Benutzer verschiedene Spielmodi an, indem eine Auswahlfenster für die Spieloptionen geöffnet wird. Die Modi umfassen "Einfach", "Mittel", "Schwer" und "Gegen Freund".

4. \*\*`open\_chess\_rules` (Zeilen 115-117, 3 Zeilen)\*\*: Öffnet einen Webbrowser mit den Schachregeln.

5. \*\*`open\_documentation` (Zeilen 119-129, 11 Zeilen)\*\*: Öffnet eine lokal gespeicherte Dokumentation in Microsoft Word. Diese Methode behandelt außerdem mögliche Fehler bei der Dokumentenöffnung.

6. \*\*`update\_bot\_information` (Zeilen 131-148, 18 Zeilen)\*\*: Aktualisiert die Bot-Informationen basierend auf dem ausgewählten Schwierigkeitsgrad. Dies umfasst das Festlegen der ELO-Wertung und der Anzeige im GUI.

7. \*\*`start\_game` (Zeilen 150-180, 31 Zeilen)\*\*: Startet ein Spiel mit dem ausgewählten Schwierigkeitsgrad. Die Methode setzt den Zustand der GUI, aktualisiert die Gegnerinformationen und startet die Timer, falls ein zeitbegrenztes Spiel gewählt wurde.

8. \*\*`start\_friend\_game` (Zeilen 182-196, 15 Zeilen)\*\*: Startet ein Spiel gegen einen Freund. Hier werden die Timer entsprechend der Spielmodus-Einstellungen initialisiert.

9. \*\*`load\_engine` (Zeilen 198-214, 17 Zeilen)\*\*: Lädt die Schach-Engine (Stockfish) und informiert den Benutzer, ob die Engine erfolgreich geladen wurde oder ob es zu Fehlern gekommen ist.

10. \*\*`browse\_for\_engine` (Zeilen 216-219, 4 Zeilen)\*\*: Bietet dem Benutzer die Möglichkeit, den Pfad zur Stockfish-Engine zu durchsuchen und auszuwählen.

11. \*\*`create\_game\_interface` (Zeilen 221-351, 131 Zeilen)\*\*: Erstellt die gesamte Benutzeroberfläche des Spiels, einschließlich des Schachbretts, der Spielfeldelemente, der Modus-Auswahl und der Timer-Anzeigen. Dies ist die längste Methode im Programm und umfasst die Gestaltung der Interaktivität des Spielbretts und die Anzeige von relevanten Informationen.

12. \*\*`update\_advantage` (Zeilen 353-368, 16 Zeilen)\*\*: Berechnet und zeigt den Materialvorteil an, der sich aus den eroberten gegnerischen Figuren ergibt. Diese Methode aktualisiert die GUI, um den aktuellen Status anzuzeigen.

13. \*\*`toggle\_clock` (Zeilen 370-382, 13 Zeilen)\*\*: Aktiviert oder deaktiviert die Zeitsteuerung des Spiels durch Klicken auf das Uhrensymbol.

14. \*\*`update\_board` (Zeilen 384-416, 33 Zeilen)\*\*: Aktualisiert das visuelle Erscheinungsbild des Schachbretts, einschließlich der Positionen der Figuren, des Schachmatt-Zustands und der möglichen Züge.

15. \*\*`update\_piece\_banks` (Zeilen 418-429, 12 Zeilen)\*\*: Aktualisiert die Stückebänke (eroberte Figuren) für beide Spieler und zeigt sie in der GUI an.

16. \*\*`reset\_game` (Zeilen 431-463, 33 Zeilen)\*\*: Setzt das gesamte Spiel zurück und stellt alle ursprünglichen Einstellungen und GUI-Komponenten wieder her.

17. \*\*`start\_timers` (Zeilen 465-471, 7 Zeilen)\*\*: Startet die Timer für den Spieler oder den Bot, je nach dem, wer an der Reihe ist.

18. \*\*`start\_player\_timer` (Zeilen 473-483, 11 Zeilen)\*\*: Zählt die verbleibende Zeit des Spielers herunter und beendet das Spiel, wenn die Zeit abläuft.

19. \*\*`start\_bot\_timer` (Zeilen 485-495, 11 Zeilen)\*\*: Zählt die verbleibende Zeit des Bots herunter und beendet das Spiel, wenn die Zeit abläuft.

20. \*\*`update\_timer\_display` (Zeilen 497-509, 13 Zeilen)\*\*: Aktualisiert die Anzeige der verbleibenden Zeit auf den Uhren für Spieler und Bot.

21. \*\*`set\_timer` (Zeilen 511-513, 3 Zeilen)\*\*: Platzhalter-Funktion zum Einstellen des Timers, derzeit leer.

22. \*\*`handle\_square\_click` (Zeilen 515-552, 38 Zeilen)\*\*: Handhabt die Benutzeraktion, wenn ein Spielfeld angeklickt wird, einschließlich der Auswahl und des Ziehens von Figuren sowie der Ausführung von Zügen.

23. \*\*`open\_promotion\_window` (Zeilen 554-589, 36 Zeilen)\*\*: Öffnet ein Fenster, um eine Bauernumwandlung auszuwählen, wenn ein Bauer die gegnerische Grundlinie erreicht.

24. \*\*`push\_move` (Zeilen 591-611, 21 Zeilen)\*\*: Führt einen Zug aus, aktualisiert das Brett, speichert eroberte Figuren und überprüft den Spielstatus.

25. \*\*`make\_move` (Zeilen 613-617, 5 Zeilen)\*\*: Aktualisiert das Schachbrett und die Anzeige nach einem Zug.

26. \*\*`switch\_timer` (Zeilen 619-624, 6 Zeilen)\*\*: Schaltet die Timer zwischen Spieler und Bot hin und her, je nachdem, wer an der Reihe ist.

27. \*\*`highlight\_moves` (Zeilen 626-633, 8 Zeilen)\*\*: Hebt mögliche Züge für eine ausgewählte Figur hervor.

28. \*\*`bot\_move` (Zeilen 635-648, 14 Zeilen)\*\*: Führt einen Zug des Bots aus, einschließlich der Behandlung von Umwandlungen und der Aktualisierung der GUI.

29. \*\*`display\_move\_in\_output` (Zeilen 650-691, 42 Zeilen)\*\*: Zeigt die durchgeführten Züge in der GUI-Ausgabe an, einschließlich der Formatierung für Rochaden, Umwandlungen und Schach.

30. \*\*`disable\_board` (Zeilen 693-698, 6 Zeilen)\*\*: Deaktiviert alle Spielfelder auf dem Schachbrett.

31. \*\*`enable\_board` (Zeilen 700-706, 7 Zeilen)\*\*: Aktiviert alle Spielfelder auf dem Schachbrett, sodass Züge möglich sind.

32. \*\*`check\_end\_game` (Zeilen 708-727, 20 Zeilen)\*\*: Überprüft auf verschiedene Spielendbedingungen (z.B. Schachmatt, Patt) und zeigt eine entsprechende Nachricht an.

33. \*\*`display\_end\_game\_popup` (Zeilen 729-747, 19 Zeilen)\*\*: Zeigt ein Popup an, wenn das Spiel endet, einschließlich der Möglichkeit, ein neues Spiel zu starten oder eine Revanche zu spielen.

34. \*\*`clear\_end\_game\_displayed` (Zeilen 749-751, 3 Zeilen)\*\*: Setzt das Flag zurück, das anzeigt, ob das Ende des Spiels bereits angezeigt wurde.

35. \*\*`reset\_piece\_banks` (Zeilen 753-756, 4 Zeilen)\*\*: Setzt die Bänke für eroberte Figuren zurück.

36. \*\*`clear\_screen` (Zeilen 758-761, 4 Zeilen)\*\*: Löscht alle GUI-Elemente auf dem Bildschirm.

37. \*\*`on\_closing` (Zeilen 763-768, 6 Zeilen)\*\*: Handhabt das Ereignis, wenn das Fenster geschlossen wird, und beendet die Schach-Engine ordnungsgemäß.

Insgesamt ist der Code in verschiedene methodische Funktionen unterteilt, die alle notwendigen Aufgaben übernehmen, um das Spiel interaktiv und benutzerfreundlich zu gestalten. Jede Methode ist auf eine spezifische Aufgabe ausgerichtet, wodurch der Code lesbarer und leichter wartbar bleibt.