Darba atskaite

# Board Class Funkcijas

1. **\_\_init\_\_(self, screen, clock, mode, winnerFont, subtitleFont):** Inicializē jaunu laukuma objektu ar norādītajiem argumentiem.
2. **initBoard(self):** Inicializē spēles laukumu ar dambretes figūriņām. Metode izveido sarakstu ar sarakstu, kas attēlo laukumu, un aizpilda to ar gabala objektiem vai nullēm, kas apzīmē tukšus kvadrātus. Metode atgriež laukumu.
3. **drawValidMoves(self, selectedPiece):** Uzzīmē izvēlētajai figūrai derīgās kustības uz spēles galda. Metode ņem kortei SelectPiece, kas attēlo atlasītā gabala koordinātas, un izmanto metodi getValidMoves(), lai noteiktu derīgās kustības. Pēc tam metode uz ekrāna derīgajās pārvietošanās pozīcijās uzzīmē zilus apļus.
4. **selectPiece(self, pieceCord):** Iestata atlasīto gabalu uz jaunu gabalu, ko nosaka rindu un kolonnu koordinātu pieceCord virkne.
5. **pieceSelectManager(self):** Apstrādā spēlētāja spēles figūras izvēli. Metode iegūst pašreizējo peles pozīciju un aprēķina laukumu kvadrāta koordinātas. Ja izvēlētajā laukumā ir figūra, kas pieder pašreizējam spēlētājam, metode atlasa figūru.
6. **drawPieces(self):** Uzzīmē visas figūras uz tāfeles. Metode veic cilpas caur katru tāfeles kvadrātu un izsauc metodi update() objektā Piece šajā vietā.
7. **drawBoard(self):** Uzzīmē spēles laukumu uz ekrāna. Metode izsauc funkciju drawSquares() no util moduļa un pēc tam izsauc metodi drawPieces(), lai uzzīmētu visus gabalus.
8. **moveManger(self):** Pārvalda izvēlētās figūras kustību uz tāfeles. Metode iegūst pašreizējo peles pozīciju un aprēķina dēļa kvadrāta koordinātas. Ja izvēlētais gājiens ir derīgs gājiens saskaņā ar spēles noteikumiem, metode pārvieto figūru uz jauno vietu un maina gājienu. Ja spēles režīms ir “pvc” un ir datora kārta, metode izsauc datora objekta metodi getMove(), lai panāktu datora kustību.
9. **winnerMessage(self):** Ekrānā parāda uzvarētāja ziņojumu. Metode atveido uzvarētāja ziņojumu un subtitru ziņojumu, izmantojot objektus winnerFont un subtitleFont. Pēc tam metode parāda ziņojumus ekrānā un gaida, līdz atskaņotājs nospiež taustiņu M, lai atgrieztos galvenajā izvēlnē.
10. **events(self):** Apstrādā Pygame notikumus, piemēram, spēles aiziešanu.
11. **update(self):** Atjaunina spēles stāvokli un grafisko displeju. Metode izsauc notikumu() ​​metodi, lai apstrādātu visus Pygame notikumus. Pēc tam metode atjaunina dēļa stāvokli, izsaucot metodi pieceSelectManager(), drawBoard() un moveManager(). Metode izsauc metodi winnerMessage(), ja ir uzvarētājs. Visbeidzot, metode atjaunina displeju, izsaucot pygame.display.flip() un gaida pulksteņa atzīmi.

# Computer Class Funkcijas

1. **\_\_init\_\_(self, computerColor):** Inicializē jaunu datora objektu ar norādīto datora atskaņotāja krāsu un iestata minimax algoritma dziļumu uz 3.
2. **events(self):** Apstrādā Pygame notikumus, īpaši pārbauda, ​​vai logs ir aizvērts.
3. 3. isInCenter(self, pieceCord): nosaka, vai gabals atrodas tāfeles centrā, pamatojoties uz tā koordinātām.
4. **eval(self, board):** Novērtē minimax algoritma efektivitāti, ņemot vērā figūru skaitu, karaļu skaitu, kontrolē esošo figūru skaitu un figūru izvietojumu (konkrēti, vai tās atrodas dēļa centrā).
5. **writeGameTreeToFile(self, gameTree):** Ieraksta spēļu koku failā "files/gameTree.txt".
6. **minimax(self, board, depth, maximizing\_player, gameTree, alpha, beta):** Ievieš minimax algoritmu ar alfa-beta apgriešanu, lai rekursīvi novērtētu visas iespējamās kustības, atkārtojot spēlētāja figūras un derīgas kustības, izvēloties gājienu ar augstāko vai zemāko vērtību atkarībā no tā, vai tas ir attiecīgi maksimizējoša vai minimizējoša spēlētājs. Tas izmanto eval metodi, lai noteiktu katras kustības vērtību, un izmanto writeGameTreeToFile, lai ierakstītu spēles koku failā.