Projectsamenvatting

Probleemomschrijving

Ik wil een programma dat een boordstate kan evalueren. Het evalueren van het boord is bedoeld om de beste posities te vinden voor een schaak AI met behulp van een search algoritme. Het search algoritme zal ik niet implementeren, maar het evaluatie algoritme is bedoeld voor een persoon die een schaak AI wil maken. Daarom wil ik mijn broer gebruiken als contactpersoon. Hij leert op het moment schaken en dit zal hem helpen. Hiervoor wil ik een GUI hebben, waarin ik kan zien wat het algoritme denkt over een boordstate. Verder wil ik het spel ook kunnen spelen.

Eisen

- Er is een schaakboord waar we op kunnen spelen
- Het schaakboord bevat alle regels en schaakstukken van schaken.
- Er is een algoritme dat het schaakboord de hele tijd evalueert.
- Het algoritme evalueert het schaakboord gebaseerd op de positie van de schaakstukken en de waarde van de schaakstukken.
- Tijdens het spelen, toont het algoritme wat hij denkt over het boord.

Algoritme

Het gekozen algoritme is een mix van twee algoritmes. Het algoritme van Kaissa[1] en het algoritme van Rob Upcraft[2]. Uit beide papers gebruiken we het algoritme dat een boord evalueert. De reden waarom ik heb gekozen voor de algoritmes is, omdat ze allebei redelijk complex zijn. Ze hebben veel goede redenen voor het geven van punten en aftrekken van punten. Hieronder staat de evaluatie criteria.

Pawn Weight +10 (Elke pawn is 10 punten waard)
Isolated pawns -1 per (Een pawn dat geïsoleerd is van andere pawns)
Doubled pawns -1 per (Twee pawns op de zelfde column)
Pawns in center +2 (Pawn zit tussen D4 en E5)
Pawn almost promoting +5 (Pawn op de rij 7 of rij 2)

Knight Weight +30 (Elke knight is 30 punten waard)
Knight in center +2 (Knight staat tussen D4 en E5)
Knight near center +1 (Knight staat tussen C4 en F6)
Knight on the rim -1 (Knight ligt aan de buitenste kant)

Bishop Weight +30 (Elke bishop is 30 punten waard)
Bishop on long diagonal +2 (Bishop zit op de langste diagonalen)
Bishop near center +1 (Bishop staat tussen C4 en F6)
Bishop on the rim -1 (Bishop ligt aan de buitenste kant)

Rook Weight +50 (Elke rook is 50 punten waard)
Rook on seventh rank +4 (Rook staat op rij 7)
Rook on openfile +3 (Rook staat op een column zonder pawns)
Rook behind past pawn +4 (Rook staat op dezelfde column achter een past pawn)

Queen Weight +90 (Elke queen is 90 punten waard)
Queen and knight exist +2 (Er is een queen en knight)

King Weight +300 (Elke king is 300 punten waard)
King castle right lost -4 (Koning heeft castle rights verloren)
King castle made +3 (Koning heeft gecasteld)
King distance from beginrow -1 per row (Koning krijgt minpunten per rij dat hij weg is van de beginrij)

Evaluatie
Er was een groot probleem opgekomen tijdens het maken van dit project. En dat was dat ik niet goed genoeg onderzoek had gedaan over hoe je een schaakboord maakt. In mijn programma is nu bijna alles gedaan met brute force, maar ik kwam er later achter dat er een veel efficiëntere methode was om bijvoorbeeld de legale zetten te halen voor een schaakstuk. Bijna alle methodes in mijn programma zijn gedaan met behulp van brute force en soms kan dit ervoor zorgen dat het programma traag is. Voor de volgende keer zou ik dus liever eerst een beeld moeten krijgen over wat ik precies moet doen. En dan daar meer onderzoek over doen, zodat ik dat efficiënter kan maken.

[1]
Donskoy, M., Adelson-Velsky, G., & Arlazarov, V. (1974, 6 augustus). *Kaissa - Chessprogramming wiki*. Chessprogramming. https://www.chessprogramming.org/Kaissa

[2]
Upcraft, R. (2015, 1 januari). *Computer Chess: A Simple Implementation*. Robupcraft.com. http://robupcraft.com/projects/ChessPaper.pdf