

Projectsamenvatting

Probleembeschrijving

Het doel van dit project is om een functionele schaakbot te ontwikkelen die in staat is om tegen een andere echte speler te spelen. Het project moet een grafische interface hebben waarin spelers schaakstukken kunnen verplaatsen, en de bot moet in staat zijn om redelijke tegenzetten te doen op basis van de huidige staat van het schaakbord. De belangrijkste uitdaging is om een algoritme te implementeren dat efficiënt en effectief beslissingen kan nemen, zelfs met beperkte rekenkracht en tijd.

Beschrijving van de Eisen

1. **Gebruiksvriendelijke Interface:** Een GUI waarin gebruikers schaakstukken kunnen selecteren en verplaatsen.
2. **Correcte Schaaklogica:** Het systeem moet alle schaakregels correct implementeren, inclusief rokade, en passant, promotie en schaakmat.
3. **AI Competentie:** De schaakbot moet in staat zijn om op een redelijk niveau te spelen, geschikt voor casual spelers.

Gekozen Algoritme

Voor dit project is het minimax-algoritme met alpha-beta pruning gekozen om de zetten te maken met een evaluatiefunctie op basis van schaakstrategieën. Het minimax-algoritme is een bekende techniek in schaakprogramma's en andere strategische spellen. Alpha-beta pruning verbetert de efficiëntie van het algoritme door takken van de zoekboom te elimineren die niet tot de beste zet kunnen leiden.

Referentie naar de bron:

- Mijn opdrachtgever Daniel die goede input heeft gegeven zodat ik de juiste richting op werkte.
- Chess Programming Wiki. "Perft Results." van [Chess Programming](#)
- Wikipedia. "Chess." van [Wikipedia](#)
- Wikipedia. "Minimax." van [Wikipedia](#)
- Wikipedia. "Alpha-beta Pruning." van [Wikipedia](#)

Onderbouwing van de Gemaakte Keuzes

4. **Minimax-algoritme:** Het minimax-algoritme werd gekozen omdat het goed is voor spellen zoals schaak. Het vindt de beste zet door alle mogelijke zetten te evalueren en de uitkomsten te vergelijken.
5. **Alpha-beta Pruning:** Door alpha-beta pruning toe te passen, kan de efficiëntie van het minimax-algoritme aanzienlijk worden verbeterd. Dit is essentieel om de rekentijd binnen acceptabele grenzen te houden, zodat de bot snel kan reageren, zelfs bij diepere zoekdiepten.
6. **Python:** Python werd gekozen als programmeertaal vanwege zijn leesbaarheid. Daarnaast heb ik dit jaar veel met Python gewerkt.
7. **Board Representatie en Evaluatie:** Het gebruik van piece-square tabellen en eenvoudige evaluatiefuncties helpt bij het snel inschatten van de bordpositie, wat essentieel is voor de efficiëntie van het algoritme. De evaluatie houdt rekening met materiële balans, controle van het centrum en ontwikkelingsvoordelen, die allemaal belangrijke factoren zijn in schaakstrategie.

Door deze keuzes te maken, hoop ik een schaakbot te ontwikkelen die functioneel is en goed voor casual spelers om tegen te spelen.