CONNECTA 4 - PROP

Mariona Farré Tapias i Pau Alcázar Perdomo

Algorisme

L'algorisme utilitzat en la classe `MinimaxPlayer` és una adaptació del clàssic algorisme Minimax amb una incorporació de la tècnica de poda Alpha-Beta. Aquest algorisme és essencial en jocs d'estratègia com Connect 4, on dos jugadors prenen torns per a realitzar moviments.

Algorisme Minimax

- Objectiu de l'Algorisme: El Minimax és un algorisme de decisió utilitzat en teoria de jocs i AI per a determinar el millor moviment en un joc de suma zero amb dos jugadors (com Connect 4, que és el nostre cas). El seu objectiu és minimitzar la possible pèrdua en un pitjor escenari, assumint que l'oponent també juga òptimament.
- Funcionament: L'algorisme simula tots els moviments possibles en el tauler, tant del jugador com de l'oponent, i avalua l'estat del joc en cada moviment. Aquests moviments i les seves conseqüències es representen en un arbre de joc, on cada node representa un estat del tauler.
- Estratègia de Minimax: En el seu torn, l'algorisme tria el moviment que maximitza la seva posició (maximitza els seus beneficis) assumint que l'oponent triarà el moviment que minimitza aquesta mateixa posició (minimitza les seves pèrdues). Això es realitza mitjançant una cerca recursiva en l'arbre de joc, alternant entre maximitzar i minimitzar en cada nivell de l'arbre.

Poda Alpha-Beta: Millorant l'Eficiència

Propòsit de la Poda: La poda Alpha-Beta és una optimització del Minimax que elimina la necessitat d'examinar cada possible moviment en l'arbre de joc. Redueix el nombre de nodes avaluats en l'arbre, millorant significativament l'eficiència de l'algorisme.

Alpha representa el millor valor que el jugador maximizador pot garantir en un nivell donat, mentre que Beta representa el millor valor que el jugador minimizador pot garantir. Durant la cerca, si es troba que un moviment no millorarà el resultat actual (si Alpha és major o igual a Beta), aquest camí es descarta, reduint així la quantitat de càlculs.

Implementació en MinimaxPlayer

- Moviment del jugador (`moviment`): Aquest mètode és el punt d'entrada per a l'algorisme Minimax. S'inicia quan és el torn del jugador per a moure una fitxa. Determina el millor moviment possible utilitzant la funció `calcularMinimax`.

- Càlcul Minimax (`calcularMinimax`): Aquí s'implementa la lògica de l'algorisme Minimax amb poda Alpha-Beta. Explora recursivamente cada possible moviment, alternant entre maximitzar i minimitzar, per a determinar el millor moviment possible en el tauler actual.
- Avaluació del Tauler (`avaluarTauler`): Fonamental per a Minimax, aquest mètode avalua l'estat del tauler de joc. Utilitza una heurística per a assignar un valor numèric al tauler, indicant què tan favorable és per al jugador.
- Exploració i avaluació: `mirarColumnes` i mètodes similars exploren els possibles moviments i utilitzen la poda Alpha-Beta per a descartar camins ineficaços. L'avaluació heurística del tauler és clau en cada pas, determinant la 'utilitat' de cada possible estat del joc.

La implementació de l'algorisme Minimax amb poda Alpha-Beta en `MinimaxPlayer` és un exemple sofisticat de com es poden utilitzar aquests conceptes en jocs d'estratègia. La combinació d'una estratègia de joc profunda (Minimax), eficiència de processament (poda Alpha-Beta) i una avaluació precisa de l'estat del joc (heurística) fa de `MinimaxPlayer` un competidor formidable en el joc de Conneca 4. La clau del seu èxit radica en la capacitat d'anticipar i planificar estratègicament, considerant tant els moviments òptims del jugador com les possibles respostes de l'oponent.

Heurística

L'heurística en la classe `MinimaxPlayer` juga un paper crucial en l'avaluació dels estats del tauler de Connecta 4. L'heurística és una manera de puntuar configuracions del tauler per a estimar que favorable és una situació particular per al jugador. La puntuació no es basa simplement en el nombre de fitxes en una línia, sinó en la qualitat de les posicions i les oportunitats que aquestes creen per a guanyar o evitar que l'oponent guanyi.

Implementació de l'heurística en MinimaxPlayer

- Avaluació general del Tauler (`avaluarTauler`): Aquest mètode proporciona una avaluació global del tauler, combinant avaluacions de columnes, files i diagonals. La puntuació total és la suma d'aquests components individuals, ajustada per a reflectir les possibilitats de guanyar o perdre.
- Avaluació de columnes (`avaluarColumna`): Cada columna del tauler s'avalua individualment. Es busca seqüències contigües de fitxes del mateix color i espais buits. La puntuació augmenta amb la longitud de la seqüència i la presència d'espais que permetin la seva extensió.
- Avaluació de files (`avaluarFila`): Funciona de manera similar a l'avaluació de columnes, però de forma horitzontal. S'enfoca a identificar oportunitats per a connectar quatre fitxes horitzontalment. La puntuació depèn de la presència de seqüències i espais oberts per a estendre-les.
- Avaluació de diagonals (`avaluarDiagonals`): Les diagonals són crítiques en Connect Four, ja que ofereixen camins addicionals per a guanyar. Aquesta avaluació considera tant diagonals ascendents com descendents. Igual que amb files i columnes, es puntua en funció de les seqüències de fitxes i els espais buits.

Aspectes de l'heurística

- Seqüències i espais: La presència de seqüències llargues (3 en línia, per exemple) incrementa significativament la puntuació, especialment si hi ha espai per a agregar una fitxa addicional. Les seqüències bloquejades per l'oponent es valoren menys, ja que ofereixen menys possibilitats de guanyar.
- Ponderació per posició: Les seqüències més pròximes al fons del tauler solen ser més valuoses, ja que ofereixen més opcions per a futures connexions. Les posicions centrals poden tenir una puntuació lleugerament superior a causa de la seva flexibilitat per a crear connexions en múltiples direccions.
- Amenaces i oportunitats: Una configuració que impedeix a l'oponent guanyar en el pròxim torn rep una alta puntuació negativa, assenyalant una situació crítica. Similarment, una configuració que permet guanyar en el pròxim torn rep una puntuació molt alta.

L'heurística en `MinimaxPlayer` és una combinació d'avaluacions detallades de diferents aspectes del tauler. No se centra únicament a comptar fitxes, sinó a analitzar el potencial de cada configuració en termes d'oportunitats de guanyar i bloquejar. Aquesta aproximació permet al jugador AI no sols reaccionar a les amenaces immediates sinó també planificar estratègicament per a futurs moviments, buscant maximitzar les seves possibilitats d'èxit mentre minimitza les de l'oponent.

Explicació dels mètodes de la classe

La classe `MinimaxPlayer` en Connect Four és un exemple complex de la implementació de l'algorisme Minimax amb poda Alpha-Beta. Aquesta classe utilitza diversos mètodes i funcions per a jugar de manera òptima. Explorarem cadascun d'aquests mètodes i funcions detalladament:

Constructors

- MinimaxPlayer(int profunditat):
    - Propòsit: Estableix la profunditat de cerca de l'algorisme Minimax.
    - Funcionament: Emmagatzema la profunditat proporcionada en `profunditatRecerca` i assigna un nom al jugador.
    - Importància: La profunditat determina quants moviments cap endavant considerarà l'algorisme, afectant el balanç entre rendiment i precisió.
- MinimaxPlayer(int profunditat, boolean estats):
    - Propòsit: Variant del constructor que permet configurar si es mostraran estadístiques.
    - Funcionament: Crida al constructor principal i estableix les estadístiques segons el paràmetre `estats`.
- MinimaxPlayer():
    - Propòsit: Constructor per defecte que assigna una profunditat predeterminada.

    - Funcionament: Utilitza una profunditat estàndard (per exemple, 5) per a casos en els quals no s'especifica.

Mètodes Principals

- nom():
    - Propòsit: Retorna el nom del jugador AI.
    - Funcionament: Retorna la variable `nomJugador`, que és una cadena que identifica al jugador AI.
- moviment(Tauler tauler, int colorJugador):
    - Propòsit: Realitza un moviment en el tauler.
    - Funcionament: Assigna el color del jugador i augmenta el comptador de jugades. Crida a `calcularMinimax` per a determinar la millor columna per a moure's.
    - Importància: És el mètode que interactua directament amb el joc, aplicant l'algorisme Minimax per a decidir moviments.

Minimax i avaluació de l'heurística

- calcularMinimax(Tauler t, int profunditat):
    - Propòsit: Implementa l'algorisme Minimax amb poda Alpha-Beta.
    - Funcionament: Inicia el procés de decisió del Minimax, considerant totes les jugades possibles fins a una certa profunditat.
- mirarColumnes(Tauler t, int profunditat, int col, Integer valor, int alfa, int beta):
    - Propòsit: Explora totes les possibles jugades en les columnes del tauler.
    - Funcionament: Recorre cada columna del tauler, calculant el valor de Minimax per a cada jugada possible i aplicant la poda Alpha-Beta.
- avaluarTauler(Tauler t):
    - Propòsit: Avalua l'estat actual del tauler.
    - Funcionament: Combina l'avaluació de files, columnes i diagonals per a donar una puntuació heurística al tauler.
    - Importància: Proporciona una 'puntuació' a l'estat actual del tauler, crucial per a les decisions de Minimax.

Avaluació detallada de components del Tauler

- avaluarColumna(Tauler t, int col) i avaluarFila(Tauler t, int fil):
    - Propòsit: Avaluen l'estat de les columnes i files, respectivament.
    - Funcionament: Verifiquen seqüències de fitxes i espais buits, assignant puntuacions basades en la proximitat a una situació guanyadora.
- avaluarDiagonals(Tauler t):
    - Propòsit: Avalua les possibilitats en les diagonals del tauler.
    - Funcionament: Similar a les avaluacions de files i columnes però aplicat a diagonals.

Mètodes Auxiliars i de decisió

- mirarColumnes, mirarFila, puntuacioH:
    - Propòsit: mètodes auxiliars per a calcular la puntuació heurística.
    - Funcionament: Aquests mètodes ajuden a descompondre la lògica d'avaluació heurística en parts més manejables.
- maxV i valorMinim:
    - Propòsit: Són essencials per a la lògica Minimax, determinant el millor i pitjor cas en cada nivell de l'arbre de joc.
    - Funcionament: `valorMaxim` maximitza la puntuació per al jugador AI, mentre que `valorMinim` minimitza la puntuació, representant la jugada de l'oponent.

Cada mètode i funció en `MinimaxPlayer` exerceix un paper crucial en la decisió estratègica en Connect Four. Des dels constructors que estableixen el jugador i la seva profunditat de cerca, passant pel mètode central de `moviment` que inicia el procés de presa de decisions, fins a les funcions d'avaluació heurística que puntuen el tauler, tots treballen junts per a crear un jugador AI avançat i competitiu. La integració del Minimax amb poda Alpha-Beta i la meticulosa avaluació heurística fan de `MinimaxPlayer` un exemple destacat de l'aplicació d'algorismes complexos en jocs d'estratègia.