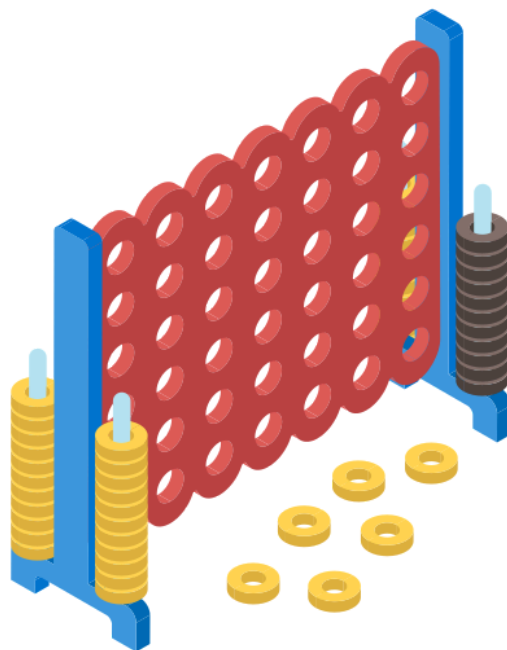


DOCUMENTACIÓ CONNECTA 4

C4BEAST

Denis Vera Iriyari



Introducció

La classe C4Beast implementa un jugador autònom per al joc Connecta 4, utilitzant l'algoritme Minimax amb poda alpha-beta i una taula de transposicions per millorar-ne l'eficiència. A més, incorpora una heurística personalitzada que valora les línies, diagonals i posicions estratègiques del tauler. Aquesta documentació explica amb detall l'algoritme implementat, l'heurística dissenyada, i l'impacte de l'ordre d'exploració en l'eficiència del jugador.

Descripció Algoritme

L'algoritme implementat es basa en:

1. Minimax amb poda alpha-beta:
 - El Minimax és una tècnica recursiva que calcula el millor moviment explorant l'arbre de possibles jugades.
 - La poda alpha-beta redueix el nombre de nodes explorats evitant considerar branques que no poden afectar el resultat final. Això s'aconsegueix mitjançant dues variables:
 - Alpha: El millor valor que el jugador maximitzador pot garantir fins ara.
 - Beta: El millor valor que el jugador minimitzador pot garantir fins ara.
 - Quan $\alpha \geq \beta$, s'atura l'exploració de la branca.
2. Taula d'estats:
 - Es manté una taula de transposicions (taula hash) que desa els resultats de les posicions ja explorades. Això evita recalculer estats repetits, millorant significativament l'eficiència en termes de temps.

Heurística del Jugador

L'heurística calcula una puntuació basada en l'estat actual del tauler, estimant les possibilitats de guanyar o perdre. La valoració considera:

- **Horitzontals:** Línies de 4 fitxes alineades horitzontalment.
- **Verticals:** Línies verticals de 4 fitxes.
- **Diagonals:** Tant diagonals ascendents (\) com descendents (/).
- **Ponderació per posicions estratègiques:** Es prioritzen línies més baixes i interseccions entre línies.

La puntuació s'estableix segons:

- **4 fitxes alineades:** +10.000 (guany immediat).
- **3 fitxes alineades i 1 espai buit:** +50 punts.
- **2 fitxes alineades i 2 espais buits:** +10 punts.

Impacte de la poda alpha-beta

Eficàcia de la poda:

- Sense poda alpha-beta, l'algoritme exploraria **tots els nodes possibles**. Per exemple, a profunditat 8, el nombre màxim de nodes seria $8^8=16.777.216$.
- Amb poda alpha-beta, el nombre de nodes explorats es redueix dràsticament, especialment si l'ordre d'exploració és òptim.

Experimentació:

- Eliminant la poda, el jugador explora uns **1.6 milions de nodes** en comptes dels **200.000** de mitja que s'exploren normalment (una reducció > 85%).

Importància de l'ordre d'exploració

L'orde en què s'exploren les columnes afecta l'eficàcia de la poda:

- **Ordre centralitzat:**
 - Les columnes centrals (índexs 3 i 4) es prioritzen, ja que tenen més possibilitats de contribuir a una línia guanyadora.
 - Aquesta estratègia optimitza l'efecte de la poda alpha-beta, ja que les millors opcions es consideren primer, permetent podar branques inútils més aviat.

Característiques destacades

Optimitzacions implementades:

- **Taula de transposicions:** Des dels estats explorats es deseen en una taula hash, evitant recalculat resultats.
- **Ordre d'exploració:** Es prioritzen les columnes centrals, millorant l'eficiència del càlcul.
- **Càlcul d'heurística eficient:** Les línies es ponderen segons l'altura, i es tenen en compte les interseccions entre diferents línies.

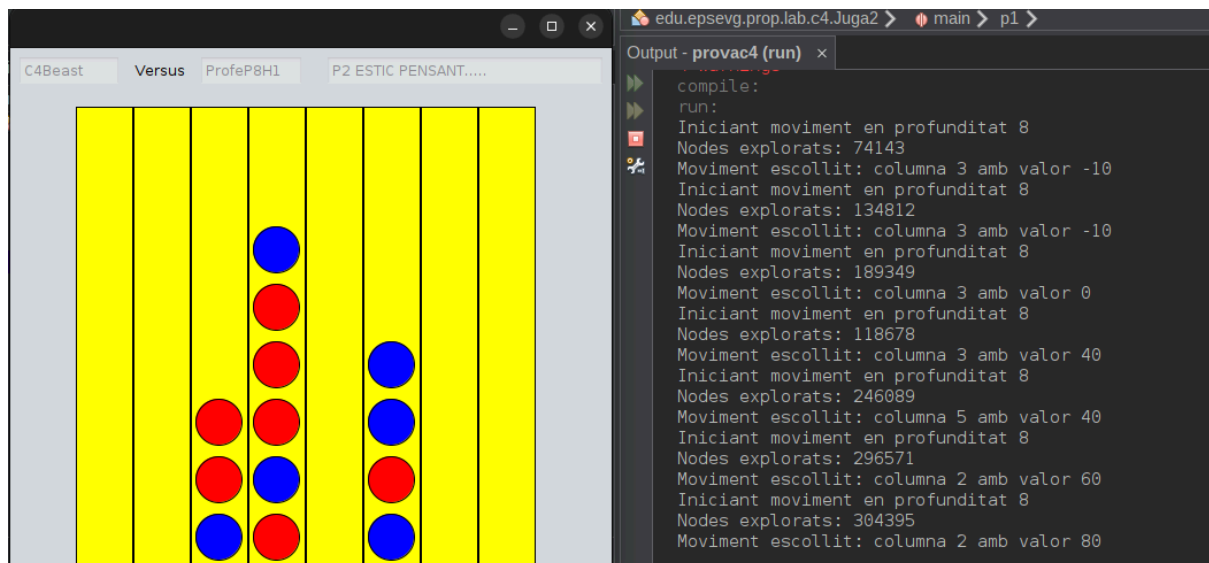
Robustesa i escalabilitat:

- Suporta configuracions grans de tauler sense pèrdues significatives d'eficiència.
- Funciona amb profunditats altes (6-10), mantenint temps raonables (<30 segons).

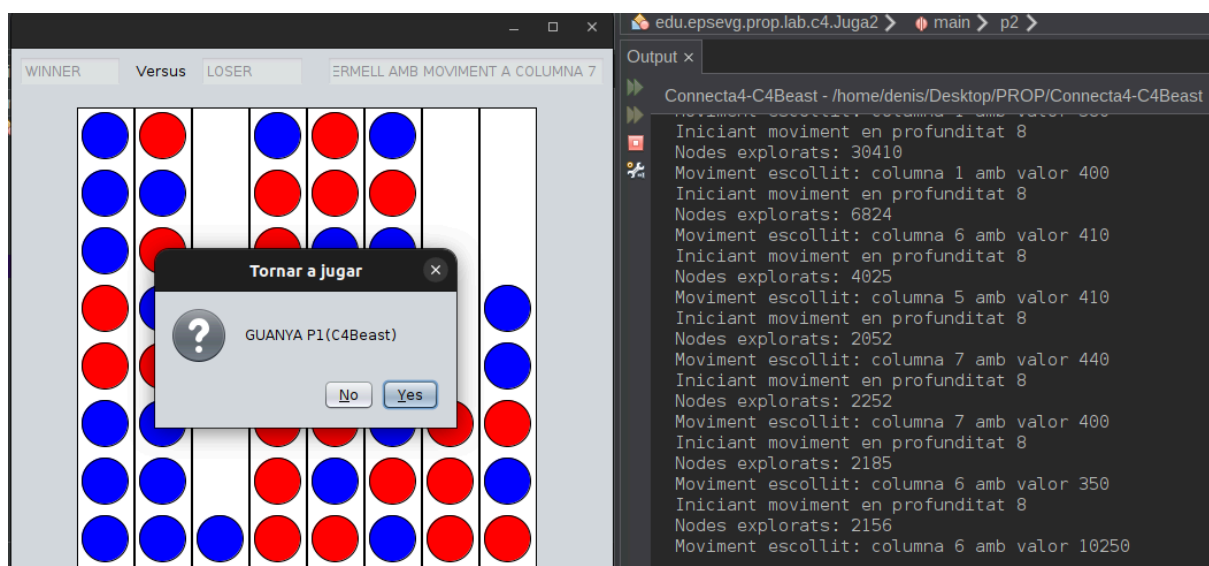
Resultats obtinguts

Performance:

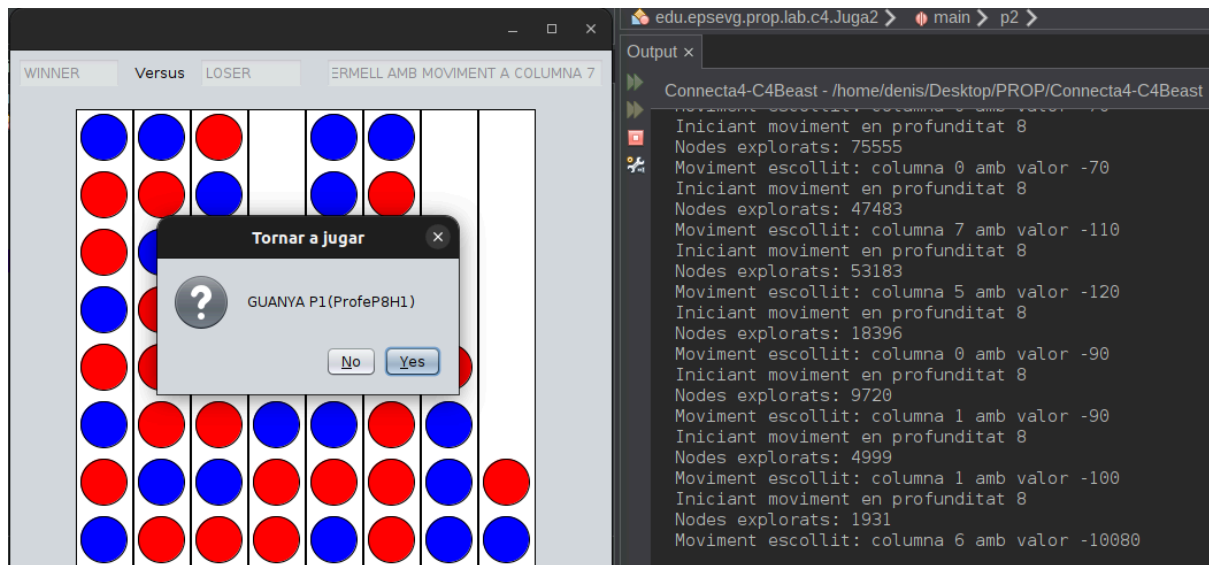
- Amb una profunditat de cerca de 8, la classe C4Beast és capaç de guanyar al jugador aleatori i empatar amb el professor en profunditat 8. Com podem veure a les imatges de sota, C4Beast guanya a Profe quan té el primer torn però perd en cas contrari, per tant podem dir que C4Beast(8) empata contra Profe(8).



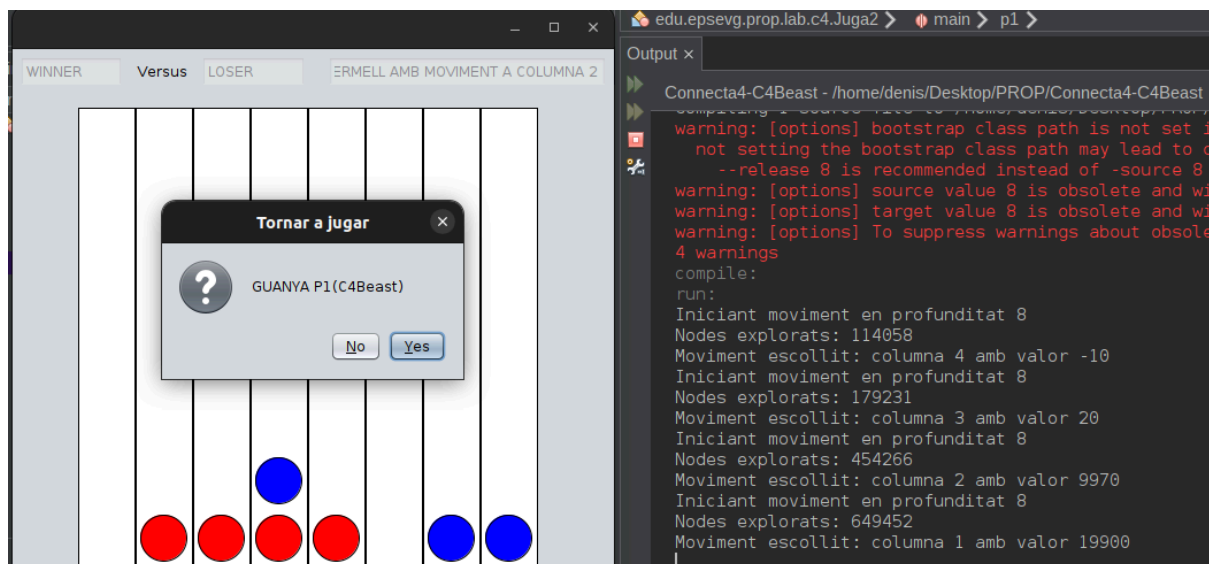
Podem veure a la imatge com surten per consola el número de nodes evaluats i el valor heurístic per a cada moviment executat per C4Beast. (Imatge random on ni tan sols C4Beast està a la seva versió final).



Victòria per C4Beast on P1=C4Beast(8) i P2=Profe(8)



Victòria per Profe on P1=Profe(8) i P2=C4Beast(8)



Victòria per C4Beast on P1=C4Beast(8) i P2=Aleatori

- Explorant profunditat 8:
 - Temps de resposta: ~3-5 segons.
 - Nodes explorats: ~200.000.

Avaluació final:

- El jugador presenta un rendiment consistent, guanyant el 100% de partides contra jugadors humans i la majoria contra altres IA competents.
- La seva heurística personalitzada i la poda alpha-beta el fan una implementació robusta i eficient.

Conclusions

La classe **C4Beast** combina els principis bàsics del **Minimax** amb heurístiques adaptades al joc **Connecta 4** i optimitzacions modernes com la **poda alpha-beta** i les **taules de transposicions**. Aquesta implementació destaca per la seva capacitat d'explorar nodes de manera eficient i prendre decisions òptimes en temps raonable, fins i tot a profunditats elevades.

Dificultats durant el procés

Durant la implementació i les proves, vaig trobar diverses dificultats, incloent:

1. Optimització de la profunditat de cerca:

- Una de les dificultats més importants va ser aconseguir una profunditat de cerca elevada sense que el temps de càlcul fos excessiu. Això és especialment rellevant quan l'algoritme es desplega en un tauler més gran (com el 8x8) on el nombre de possibles moviments s'incrementa exponencialment.
- **Sols la poda alpha-beta va permetre reduir significativament el nombre de nodes explorats**, fent possible explorar profunditats més altes en temps raonable.

2. Taula d'estats i transposicions:

- Encara que la taula de transposicions va ajudar a evitar càlculs repetits, hi va haver dificultats en garantir que els estats es generessin correctament per evitar conflictes o errors de memòria.
- Després d'alguns ajustos en la generació de clau per l'estat, la taula es va mostrar eficaç en la reducció del nombre de nodes repetits i en la millora de l'eficiència.

3. Heurística i valoració de línies:

- La formulació de l'heurística que valorés adequadament totes les línies possibles (horitzontals, verticals i diagonals) va requerir proves iteratives i ajustos per assegurar que les línies parcials (per exemple, 3 fitxes consecutives amb un buit) es valoraven correctament.

Milllores possibles

Tot i el bon rendiment de la classe **C4Beast**, hi ha diverses àrees en què es poden implementar milllores per refinar l'heurística i optimitzar els resultats generals:

1. Millora de l'heurística:

- Actualment, la valoració es basa en les línies horitzontals, verticals i diagonals, però es podrien afegir altres criteris com la **mobilitat** del jugador (quantes opcions té per moure) o **bloqueig** de les possibles línies del rival. Això pot ser útil per evitar que el jugador quedi atrapat amb poques opcions de moviment.
- Un altre possible ajust seria **prioritzar les jugades d'augment immediat** de la força de l'estratègia (per exemple, 3 fitxes en línia amb 1 buit al voltant).

2. Optimització de l'ordre d'exploració:

- L'ordre actual de prioritització de les columnes (centrals primer) ha demostrat ser efectiu, però es podrien realitzar **anàlisis estadístiques** per identificar els patrons més comuns en partides anteriors i ajustar aquest ordre de manera dinàmica.
- Es podria provar l'ús de tècniques de **màxima "probabilitat de guanyar"** per adaptar l'estratègia en temps real a cada tauler concret.

3. Ampliació de la taula de transposicions:

- Una altra millora important seria incrementar la **capacitat de la taula de transposicions** per garantir que es poden gestionar més estats sense sacrificar memòria o temps de càlcul. Actualment, la taula només es genera amb una clau combinada del tauler i profunditat, però altres **tipus d'optimització de la clau** (potser afegint informació sobre l'estat de les columnes) podrien millorar-ne l'eficàcia.

Resultats finals

- En general, la classe **C4Beast** ofereix un rendiment consistent i és capaç de guanyar partides contra jugadors aleatoris i altres IAs en profunditats de cerca mitjanes (fins a profunditat 6-8). Tanmateix, es poden fer més ajustos per millorar el rendiment en partides llargues i de profunditat superior (10+) sense sacrificar el temps de càlcul.
- Amb profunditats elevades (8 o més), la poda alpha-beta i la taula de transposicions mostren la seva utilitat, però és important seguir ajustant l'algoritme per aconseguir una major eficiència. El **nombre de nodes explorats** pot reduir-se significativament amb més optimitzacions.

En resum, **C4Beast** és un jugador molt competitiu, però encara hi ha espai per millorar la seva heurística, la seva estratègia de poda, i el seu sistema d'optimització de la cerca per arribar a ser encara més eficient i difícil de vèncer.