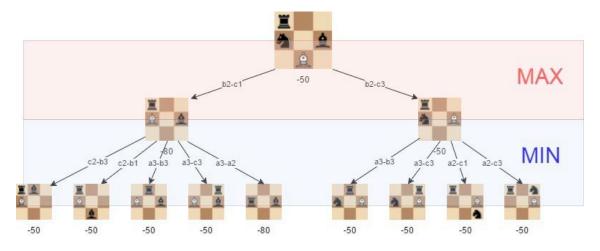
A través de la funció millorTirada cridem la funció minimax la qual requereix de 4 arguments (la profunditat, l'alpha "-10000", la beta "10000" i si vol el valor més alt o el més baix (isMaximizing)

```
public Tirada millorTirada(){
  Tirada millorTirada = null;
  Integer bestScore = 0;
  Integer score = null;
  ArrayList<Tirada> res = new ArrayList<>();
  res = obtenirTiradesPossiblesJugador(_torn);
  Boolean isMaximizing = null;
  if(_torn == 1) {
    isMaximizing = true;
    bestScore = -9999;
  else if (torn == -1) {
    isMaximizing = false;
    bestScore = 9999;
  }
  for(Tirada tirada : res){
    aplicaMoviment(tirada,true);
    score = minimax(_maxDepthAI-1,-10000,10000, isMaximizing);
    desferTirada(tirada);
    if(isMaximizing){
      if(score > bestScore){
        bestScore = score;
        millorTirada = tirada;
      }
    else if(!isMaximizing){
      if(score < bestScore){</pre>
        bestScore = score;
        millorTirada = tirada;
    }
  return millorTirada;
}
```

Amb l'ajuda de la funció obtenirTiradesPossiblesJugador() aconseguim totes les tirades possibles que pot fer el jugador que esta tirant, un cop obtingudes passem a traves de totes ells i les apliquem al tauler a través d'un for i cridem la funció minimax.

```
public Integer minimax(Integer depth, Integer alpha, Integer beta, Boolean isMaximizing) {
  if(depth == 0) return valorTauler();
  if (isMaximizing) {
    Integer maxEva = -9999;
    ArrayList<Tirada> res = obtenirTiradesPossiblesJugador(1);
    for (Tirada tirada : res) {
      aplicaMoviment(tirada, true);
      Integer score = minimax(depth - 1, alpha, beta, !isMaximizing);
      desferTirada(tirada);
      maxEva = max(maxEva,score);
      alpha = max(alpha, maxEva);
      if(beta <= alpha) return maxEva;</pre>
    return maxEva;
  else {
    Integer minEva = 9999;
    ArrayList<Tirada> res = obtenirTiradesPossiblesJugador(-1);
    for (Tirada tirada : res) {
      aplicaMoviment(tirada, true);
      Integer score2 = minimax(depth - 1, alpha, beta, !isMaximizing);
      desferTirada(tirada);
      minEva = min(minEva,score2);
      beta = min(beta, minEva);
      if(beta <= alpha) return minEva;</pre>
    return minEva;
}
```

La funció minimax tornarà a obtenir totes les jugades possibles del jugador fins que arribi a la profunditat 0 (depth), aleshores retornarà el valor del tauler i calcularà el màxim d'una tirada i el mínim de les següent i així successivament aconseguirem la jugada més optima.



Hem de suposar que si nosaltres volem aconseguir el millor valor de tauler per nosaltres (si som les blanques serà com més petit millor i si som les negres com més gran millor).

L'alpha i la beta sutilitzen per poder descartar tirades (podar l'arbre) si el resultat que n'obtindrïem serà indiferent i així fer més ràpida la recursió.